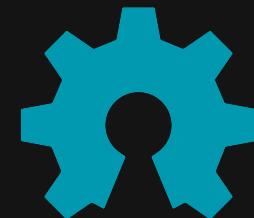
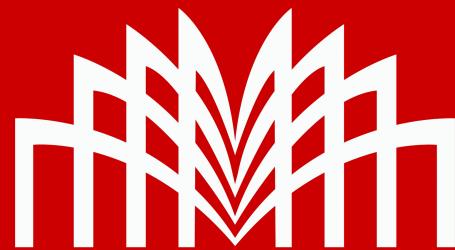


# Electronics Prototyping with Arduino



open source  
hardware



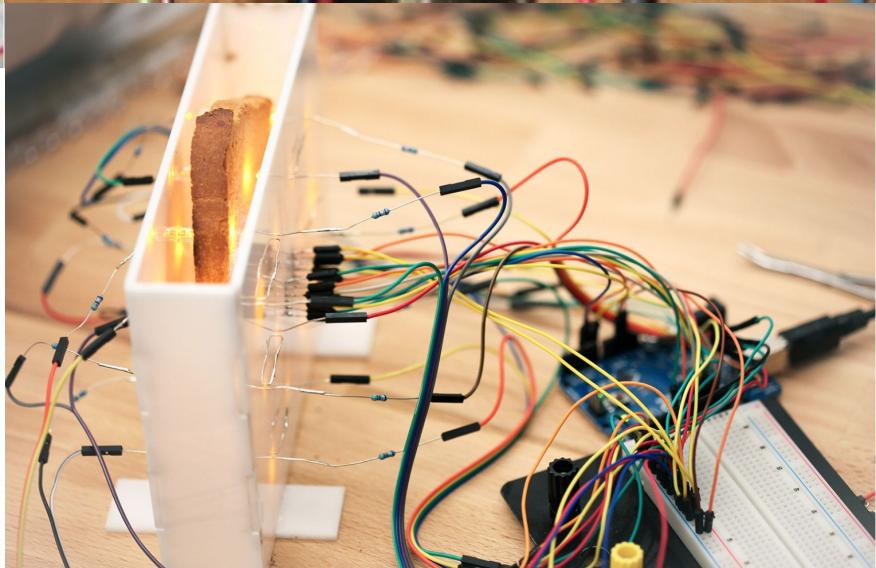
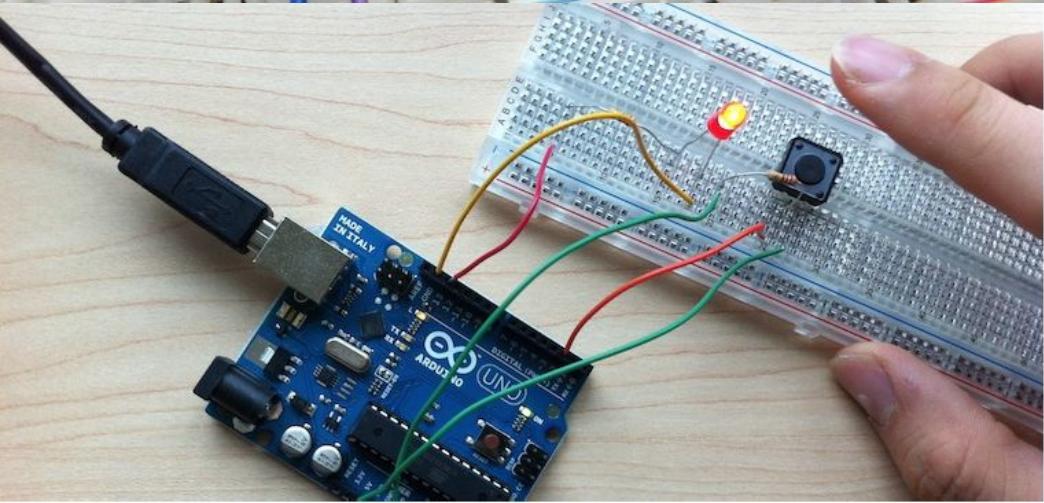
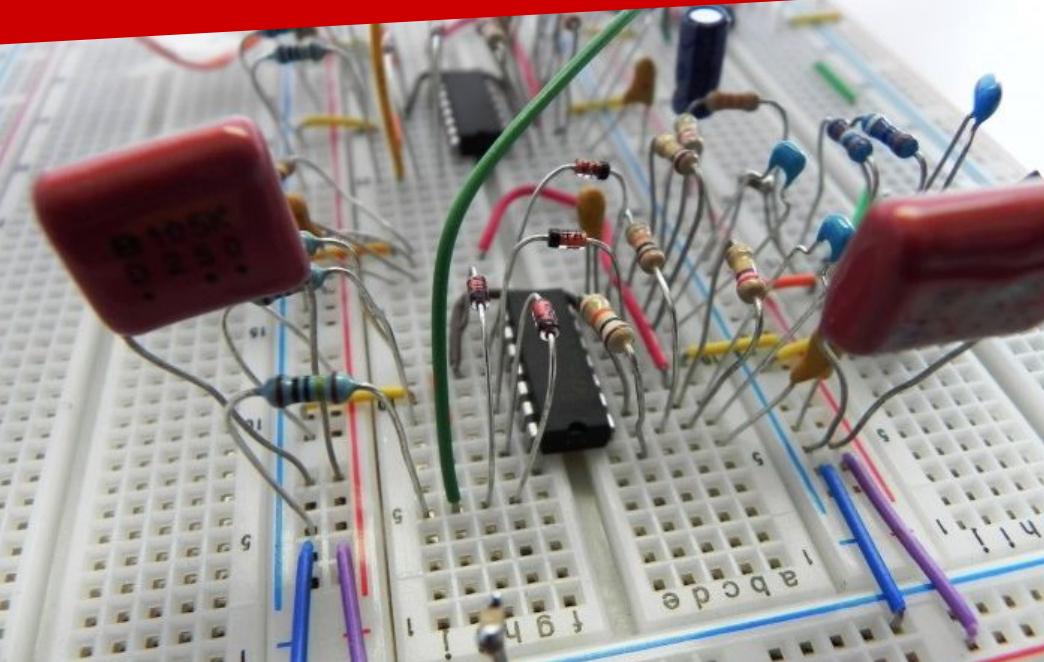
# #WorkshopGoals

1. Show the potential of electronics and the Arduino platform.
2. Build your first *working* electronics project.
3. Tinker & experiment.
4. Offer resources for building more creative and complex projects.

# Prototype

- To test a concept or process;
- A thing to be replicated or learned from.

# Electronics Prototyping



# Electronics Options

The library offers these options:



**Sparkfun Arduino Inventor Kit w/  
Wifi Shield**

- Arduino Compatible
- Easiest to get started



**Internet of Things Starter Kit**

- Arduino Compatible
- Built in Wifi and Bluetooth
- Lots of components included



**Raspberry Pi**

- Full Linux with HDMI output
- Connect via Wifi or internet



**LightBlue Bean**

- Bluetooth-connected Arduino
- Tiny and low power
- Great for wearables and mobile applications

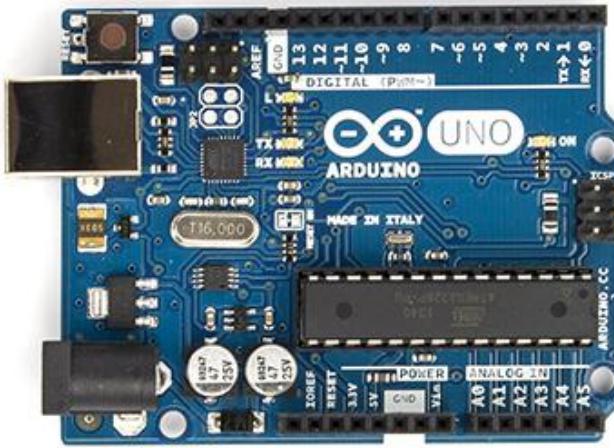
at the Ask Us desk at Hill and Hunt

# Arduino

The screenshot shows the Arduino IDE interface with the title bar "ESP8266\_Simple\_Button\_Light | Arduino 1.6.8". The code editor contains the following sketch:

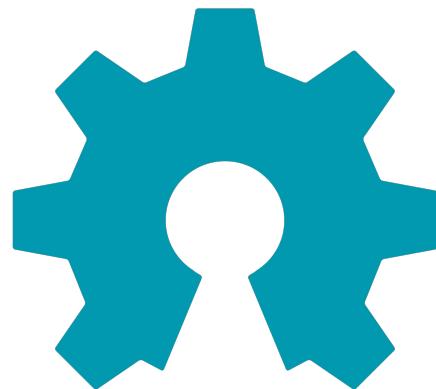
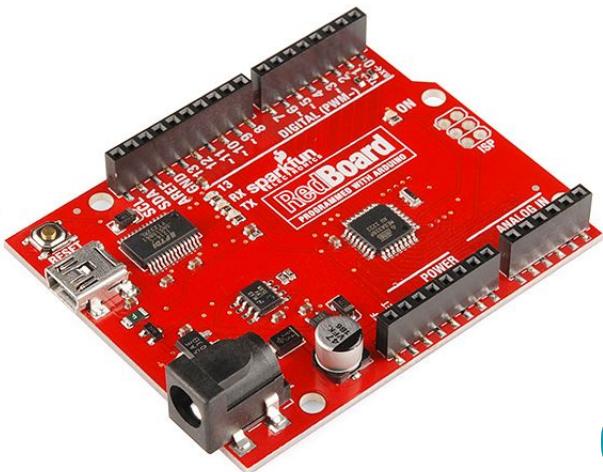
```
// constants won't change. They're used here to set pin numbers:  
const int buttonPin = 10; // the number of the pushbutton pin; note pin 10 here is pin SD3  
const int ledPin = 2; // the number of the LED pin; note pin 2 here is pin D4 on ESP8266  
  
// variables will change:  
int buttonState = 0; // variable for reading the pushbutton status  
  
void setup() {  
  // initialize the LED pin as an output:  
  pinMode(ledPin, OUTPUT);  
  // initialize the pushbutton pin as an input:  
  pinMode(buttonPin, INPUT);  
}  
  
void loop() {  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  
  // check if the pushbutton is pressed.  
  // if it is, the buttonState is HIGH:  
  if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
    delay(1000); // this will leave the light on for 1 second after the button is pressed  
  } else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
  }  
}  
  
Done uploading.  
at java.net.AbstractPlainDatagramSocketImpl.join(AbstractPlainDatagramSocketImpl.java:  
at java.net.MulticastSocket.joinGroup(MulticastSocket.java:323)  
at javax.jmdns.impl.JmDNSImpl.openMulticastSocket(JmDNSImpl.java:463)  
at javax.jmdns.impl.JmDNSImpl.<init>(JmDNSImpl.java:420)
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) on /dev/cu.wchusbserial1410

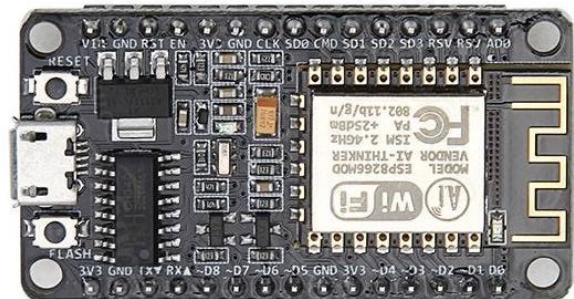
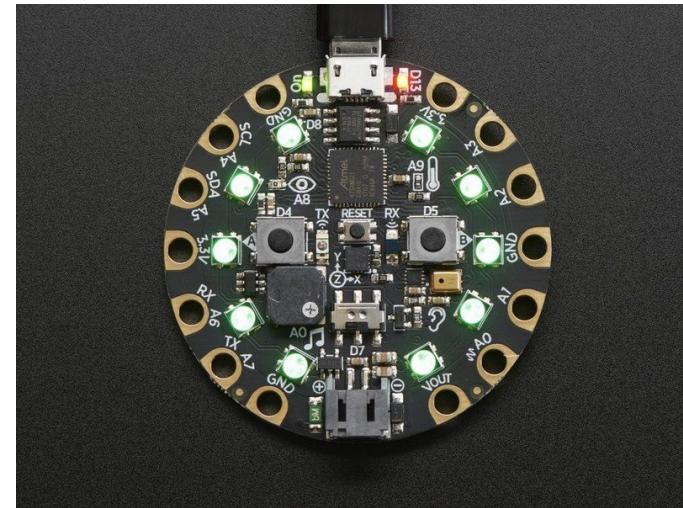
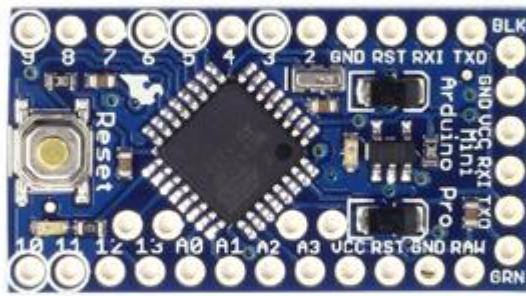
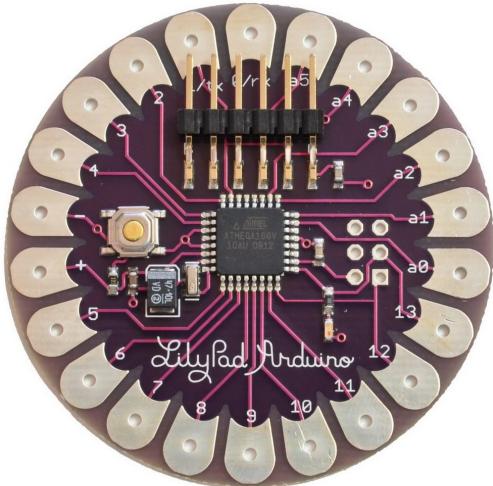


- *digital (code) + physical (electronics)*
- inputs & outputs

# (Lots of Arduinos)



open source  
hardware



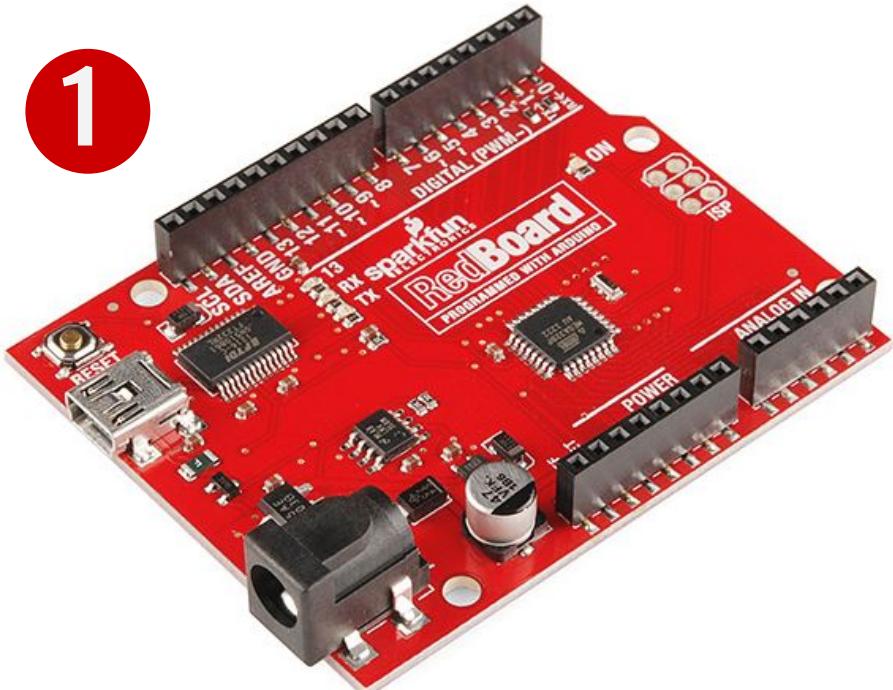
# ARDUINO

An  
Introduction  
to  
the  
Arduino



# Arduino

1



2

StandardFirmata | Arduino 1.0.4

StandardFirmata helloworld §

```
/*
Button

Turns on and off a light emitting diode(LED) connected to digital
pin 13, when pressing a pushbutton attached to pin 2.

The circuit:
* LED attached from pin 13 to ground
* pushbutton attached to pin 2 from +5V
* 10K resistor attached to pin 2 from ground

* Note: on most Arduinos there is already an LED on the board
attached to pin 13.

created 2005
by DojoDave <http://www.0j0.org>
modified 30 Aug 2011
by Tom Igoe

This example code is in the public domain.

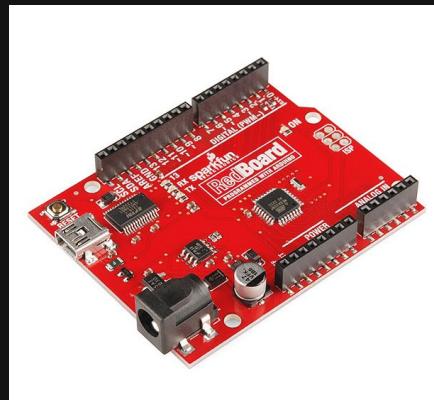
http://www.arduino.cc/en/Tutorial/Button
*/
// constants won't change. They're used here to
```

56

Arduino Uno on /dev/cu.usbmodemfd121

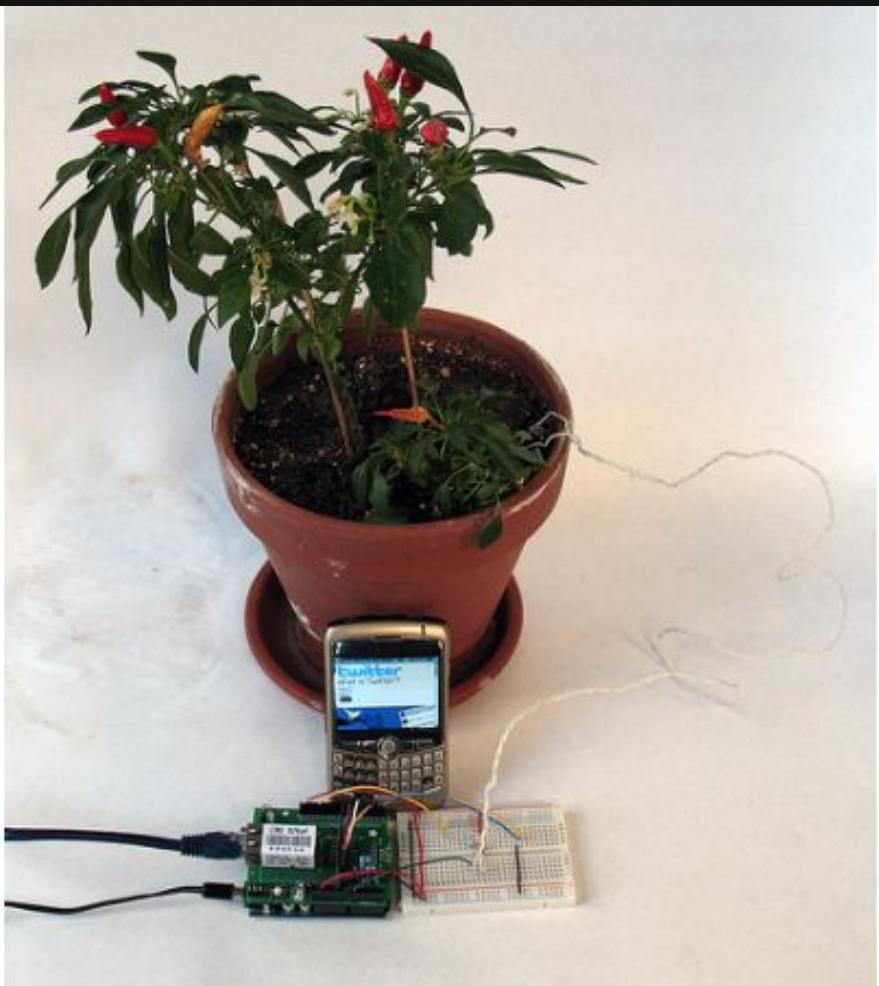
# Inputs & Outputs

data streams  
user interactions  
moisture  
human movement  
text/multimedia  
power/battery  
light



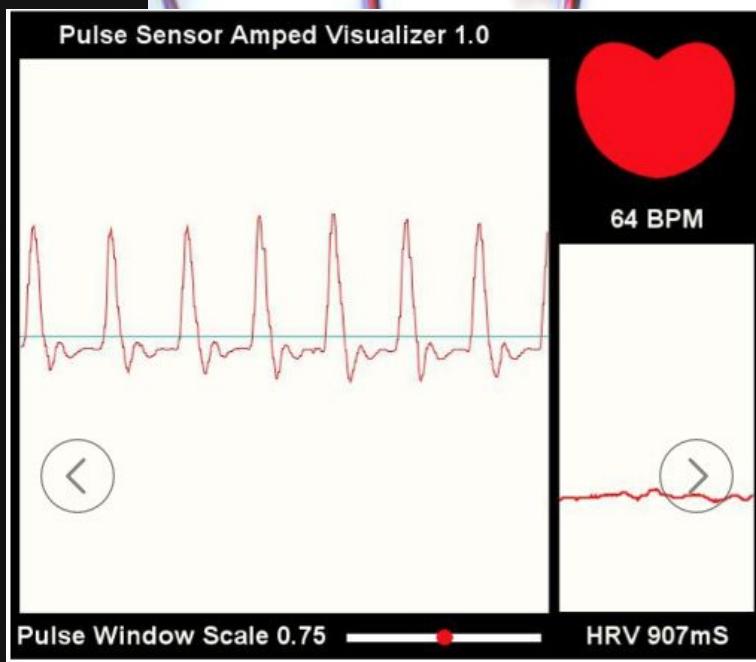
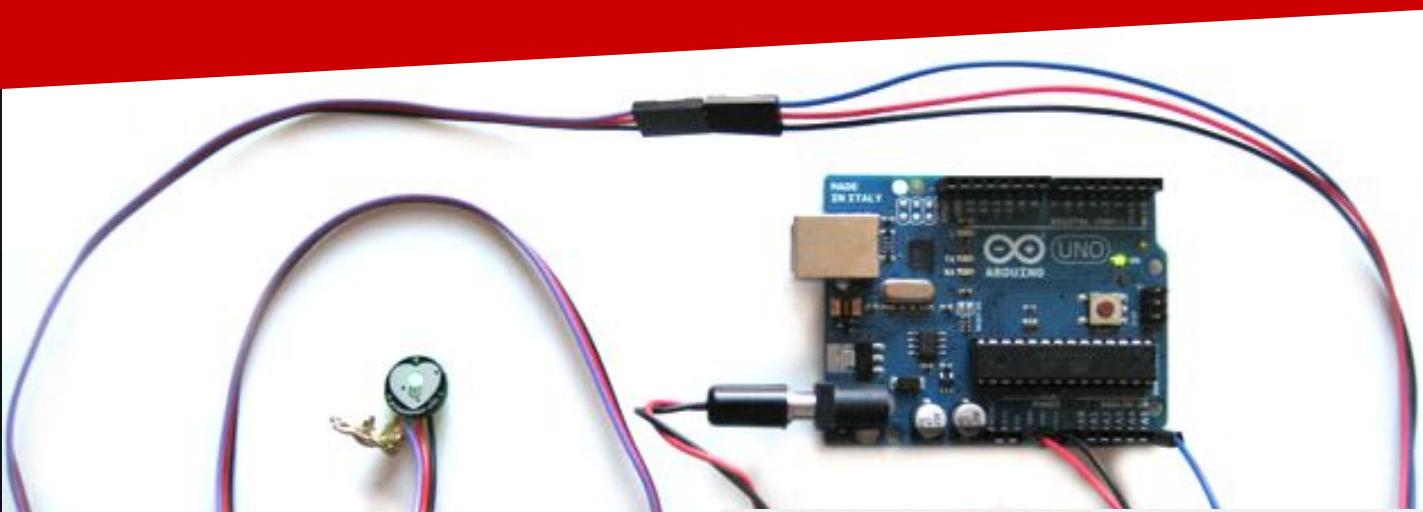
Twitter/web  
data visualization  
actuators  
sound  
light

# Examples

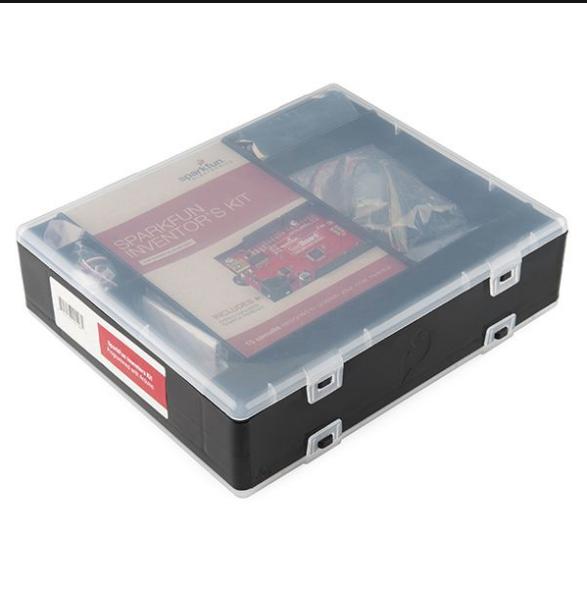


- Botanicalls** @botanicalls · 29 Oct 2008  
Thank you for watering me!
- Botanicalls** @botanicalls · 26 Oct 2008  
URGENT! Water me!
- Botanicalls** @botanicalls · 24 Oct 2008  
Water me please.
- Botanicalls** @botanicalls · 1 Oct 2008  
You over watered me.
- Botanicalls** @botanicalls · 1 Oct 2008  
Current Moisture: 64%.
- Botanicalls** @botanicalls · 11 Jun 2008  
Water me please.

# Examples



# Your Tools



# Using your own computer?

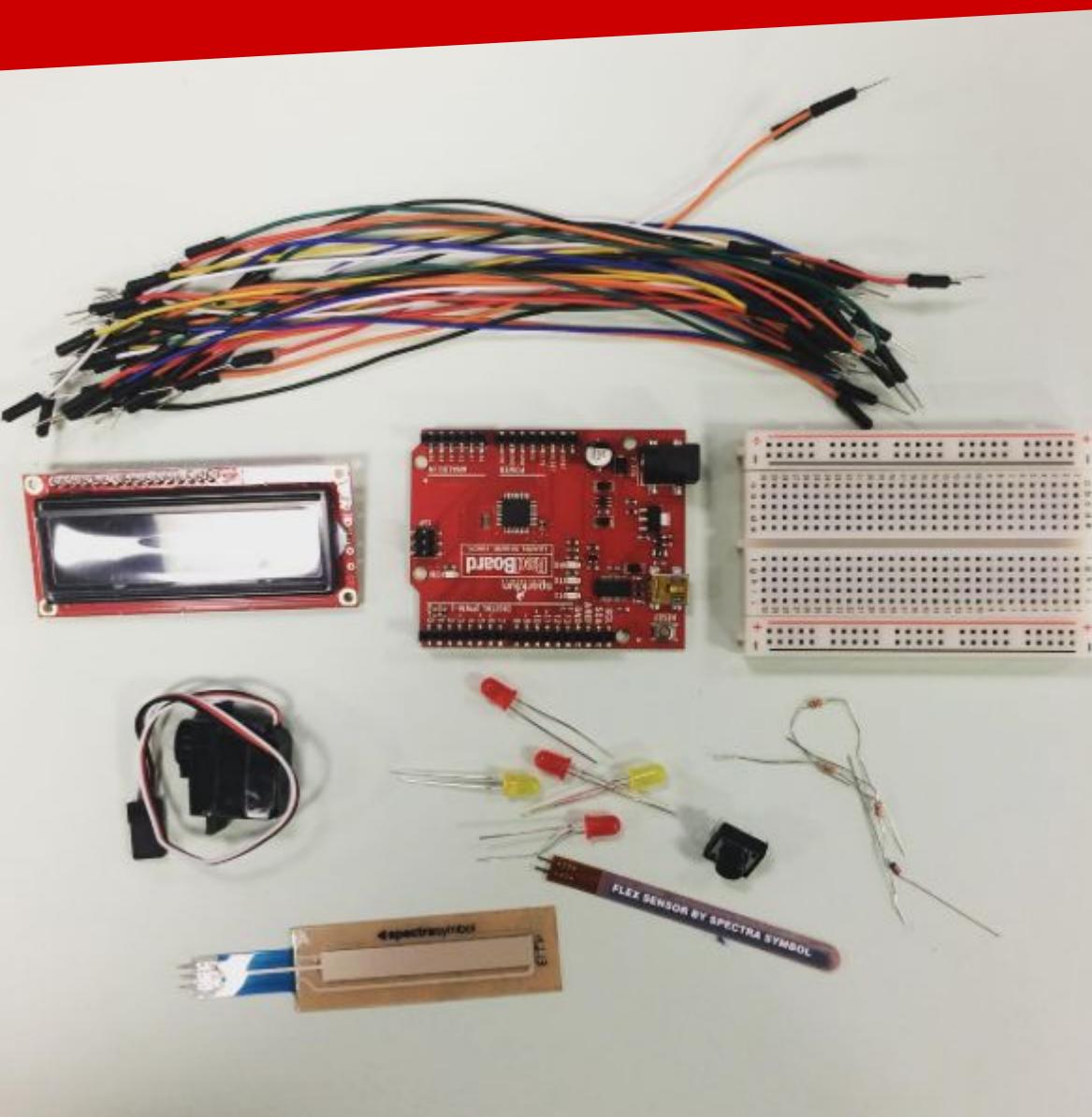
Go to: [www.arduino.cc](http://www.arduino.cc)

The screenshot shows the Arduino website's software page. At the top, there is a navigation bar with links for Home, Buy, Software, Products, Learning, Forum, Support, and Blog. An orange arrow points to the "Software" link. To the right of the navigation bar is a search bar labeled "Search the Arduino Website" and a language selection dropdown set to "ENGLISH". Below the navigation bar, the page title is "SOFTWARE". The main content area features a large image of the Arduino logo (infinity symbol with minus and plus signs) and a section titled "ARDUINO WEB EDITOR". This section explains the benefits of using the web editor, such as coding online and saving sketches in the cloud. It includes a "Try It Now" button and a "Getting Started" link. To the right of this text is a screenshot of the Arduino Web Editor interface, showing a code editor with "void setup()" and "void loop()" functions.

Download the Arduino IDE

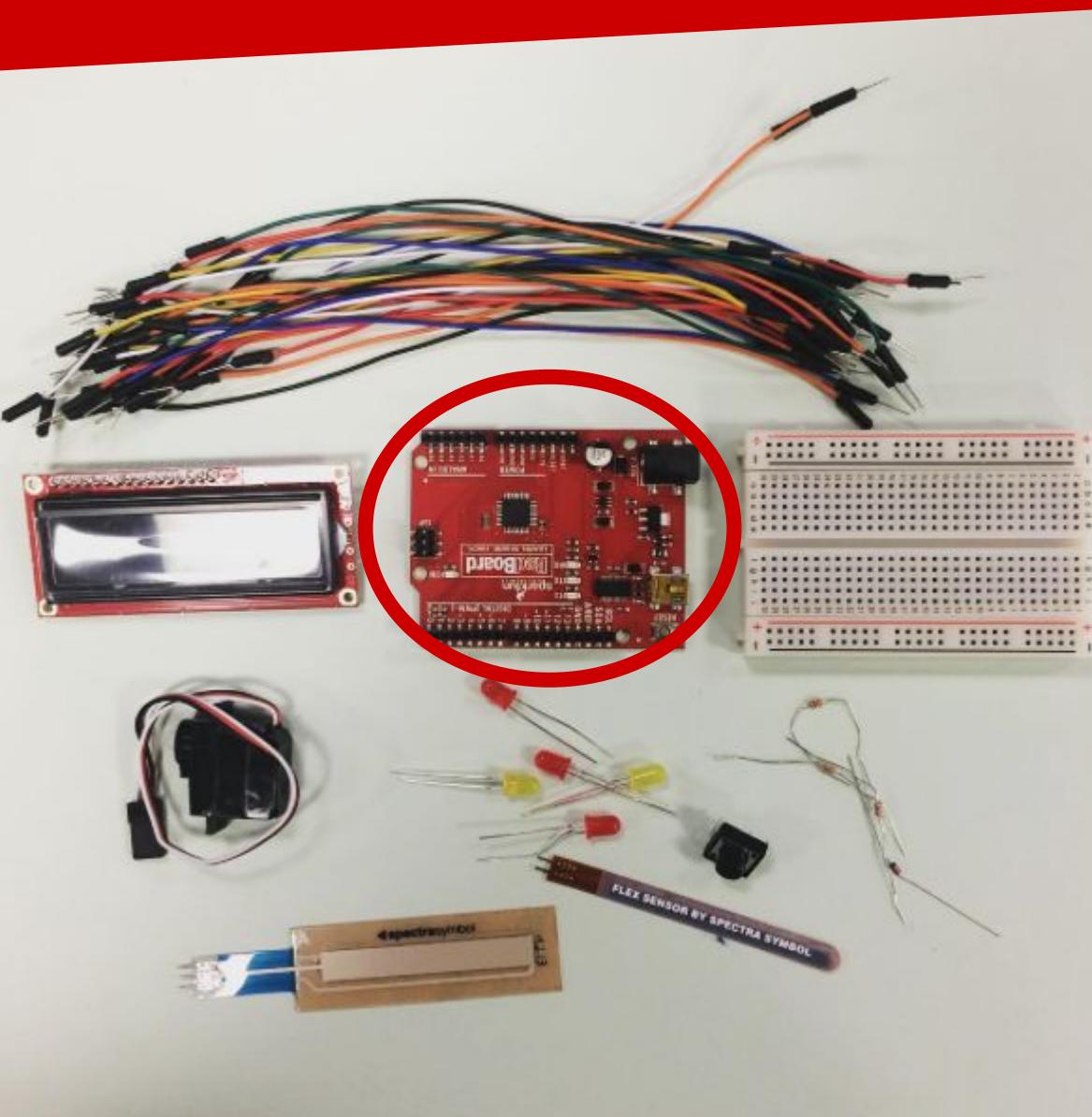
The screenshot shows the download section of the Arduino website. It features the Arduino logo at the top left. A large orange arrow points to the "ARDUINO 1.8.1" heading. Below this, there is a brief description of the Arduino Software (IDE). To the right, there are download links for different operating systems: "Windows Installer", "Windows ZIP file for non admin install", "Windows app Get", and "Mac OS X 10.7 Lion or newer". There are also links for "Linux 32 bits" and "Linux 64 bits".

# Materials



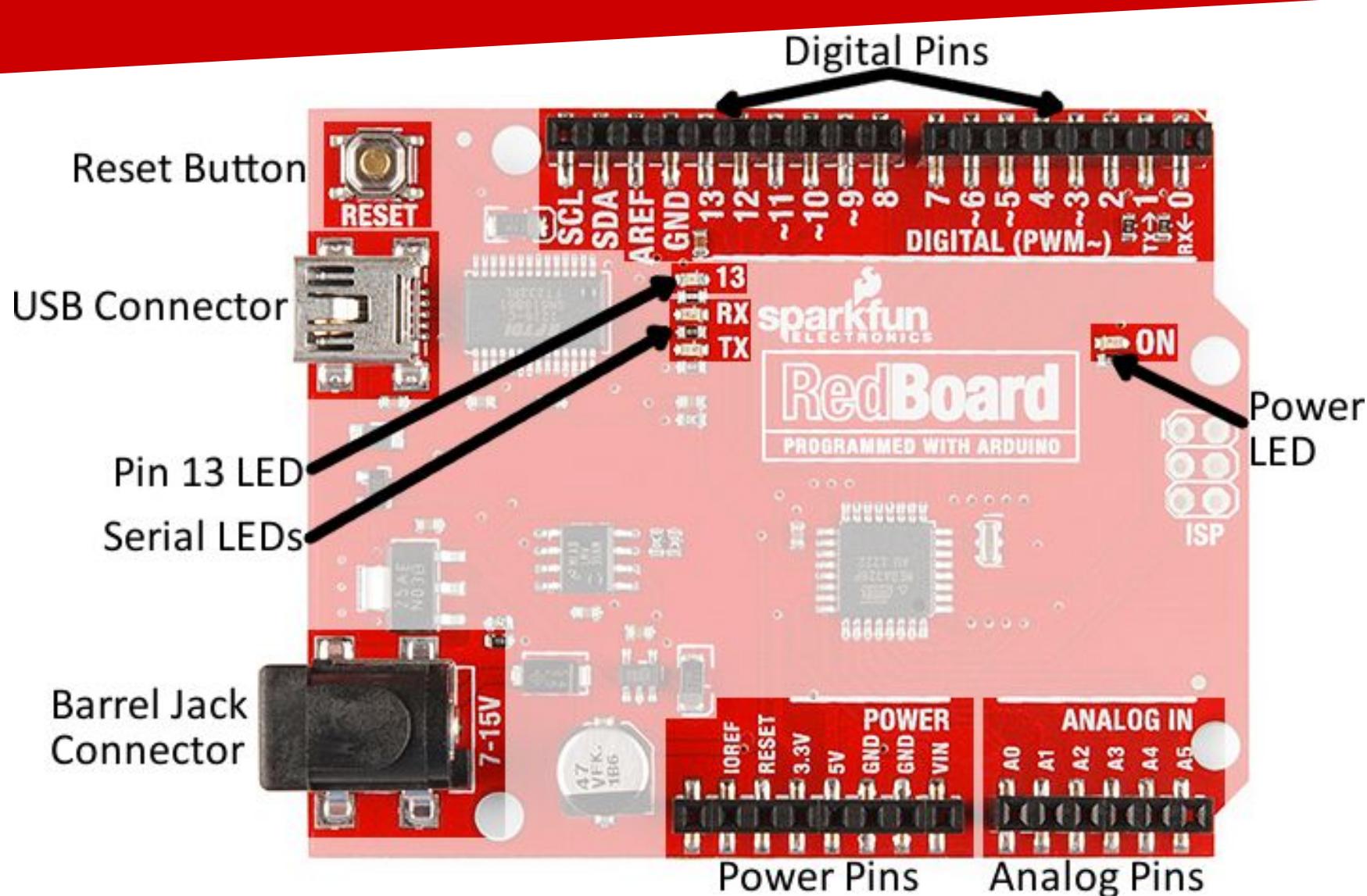
- Arduino
- Breadboard
- Components
- Jumper wires

# Materials



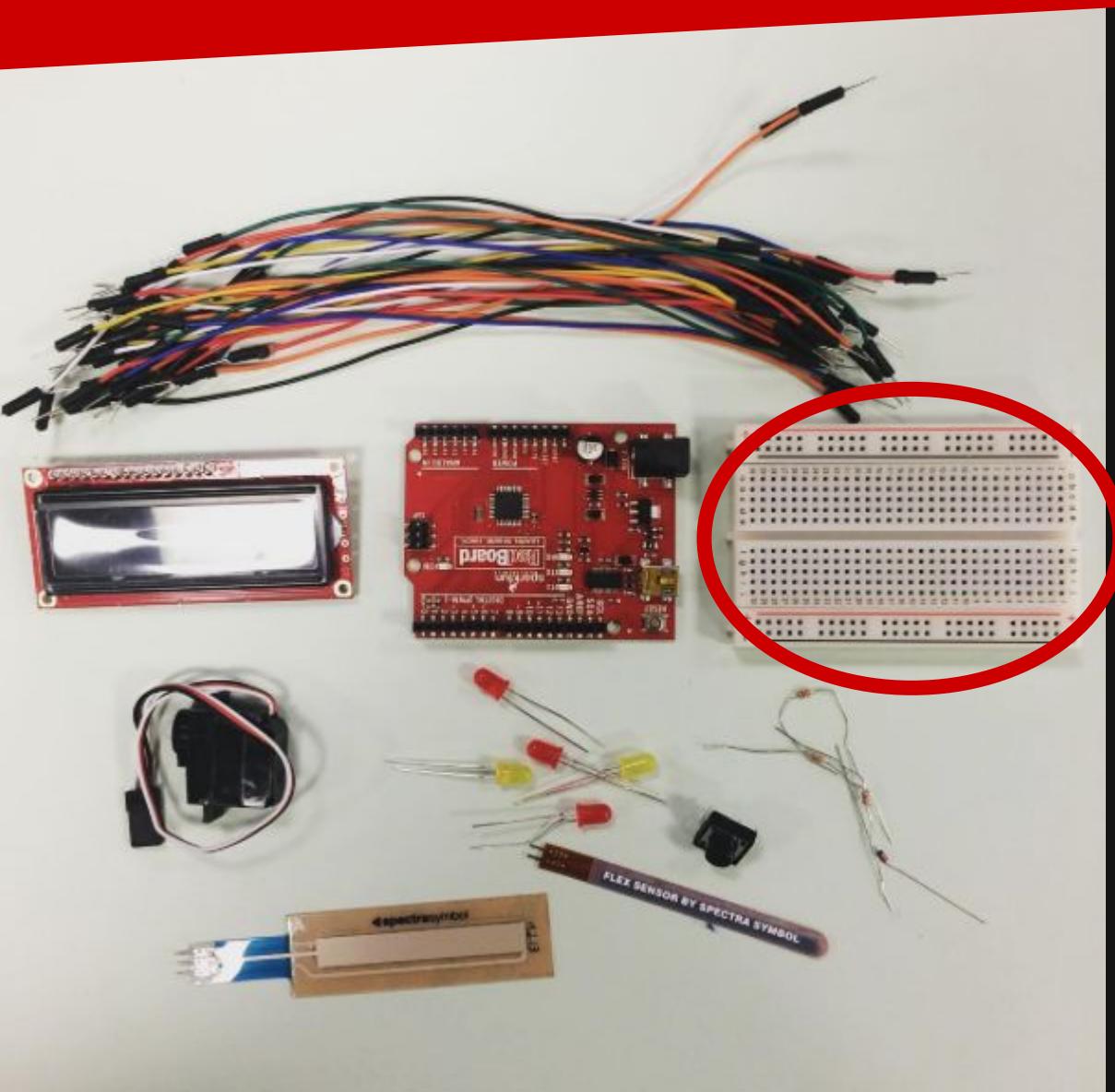
- Arduino
- Breadboard
- Components
- Jumper wires

# Arduino Anatomy



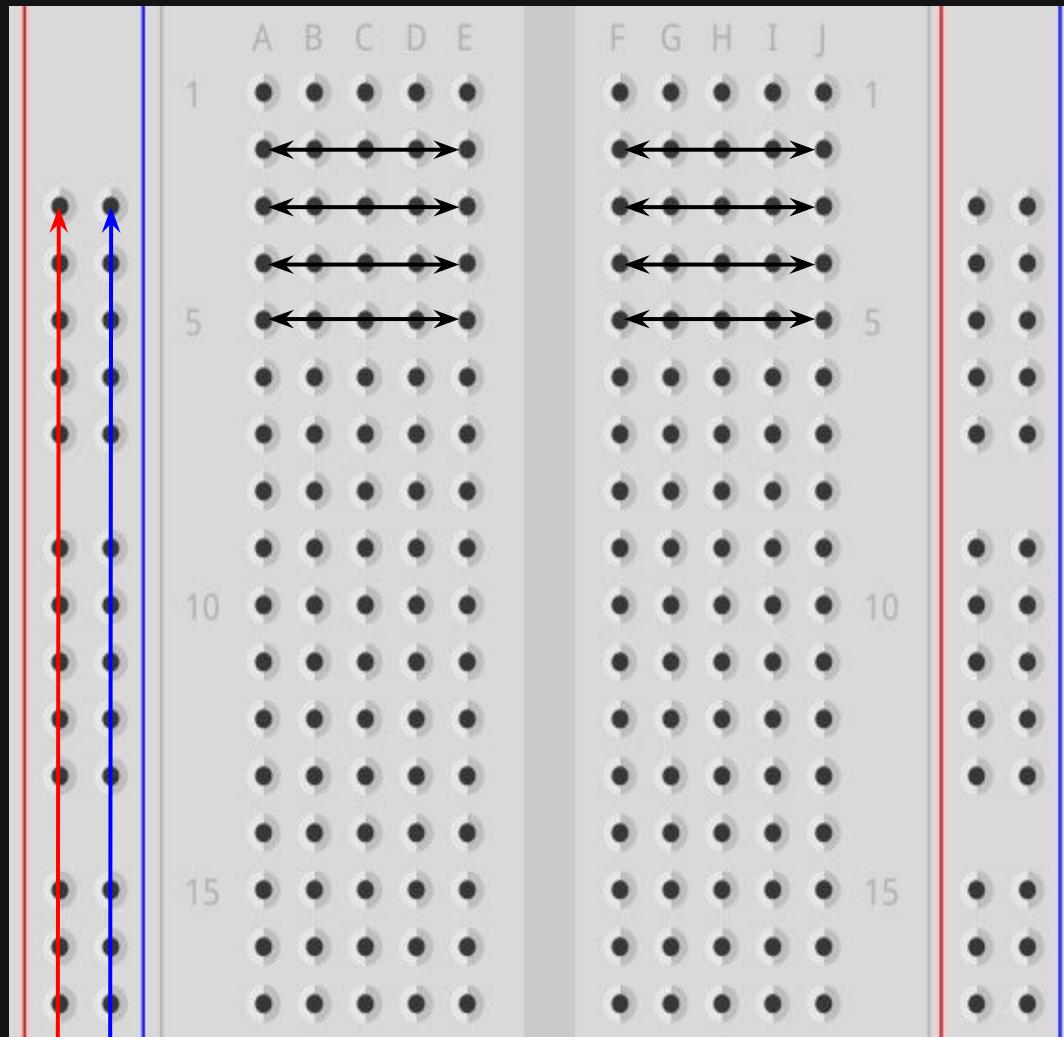
Source: <https://learn.sparkfun.com/tutorials/redboard-hookup-guide>

# Materials

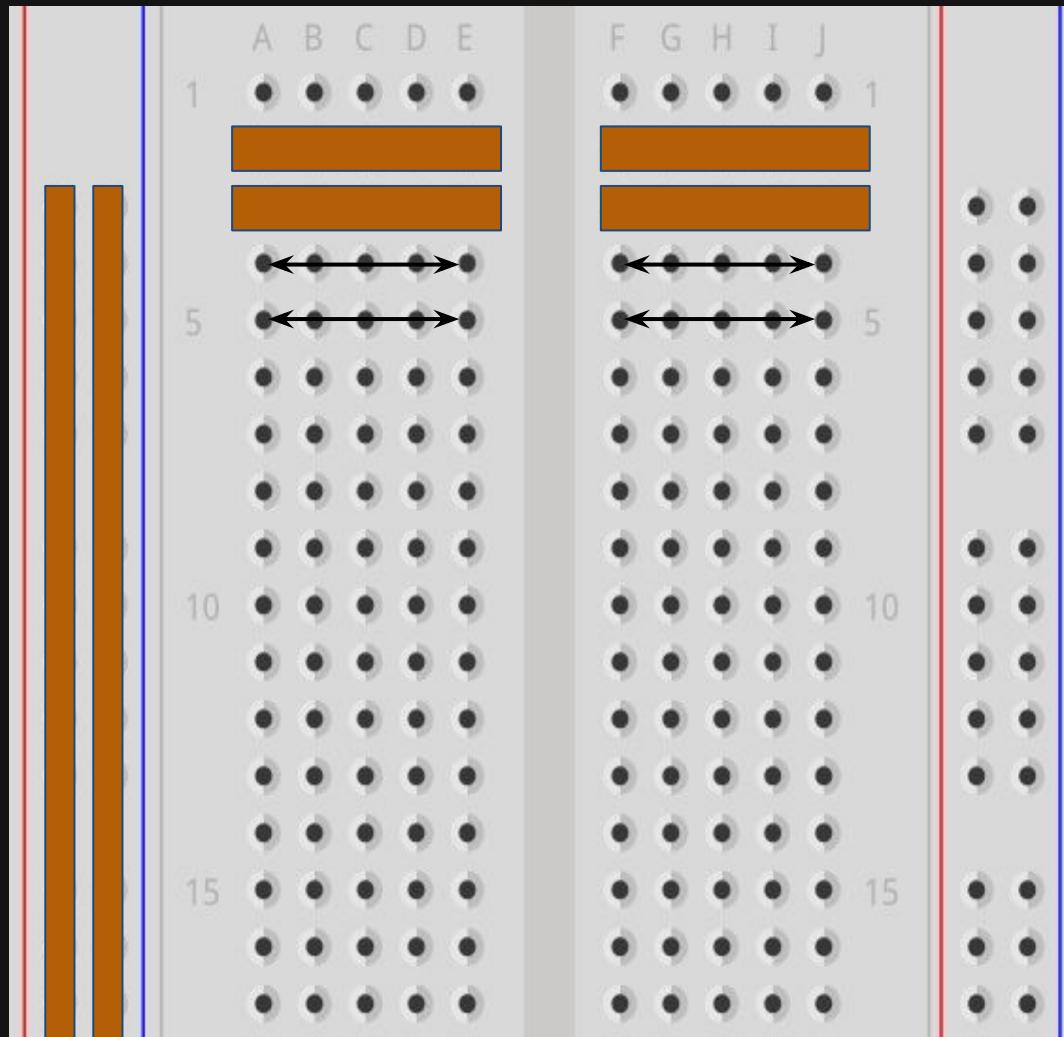


- Arduino
- Breadboard
- Components
- Jumper Wires
-

# Breadboard



# Breadboard



# Materials



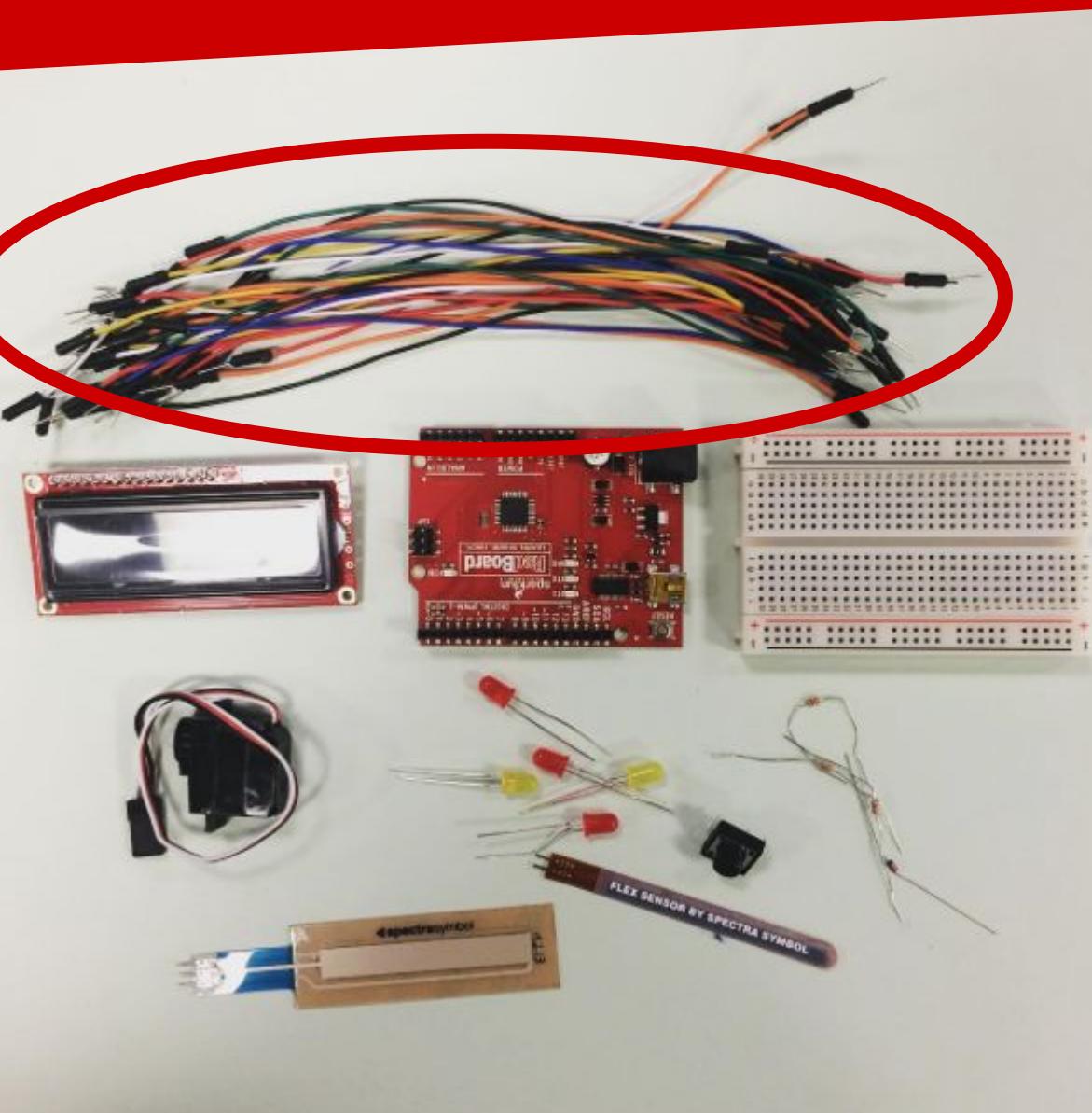
- Arduino
- Breadboard
- Components
- Jumper Wires

# Components

- Buttons
- Switches
- Sensors
- Displays
- Motors
- Resistors
- Potentiometers
- Diodes
- Capacitors

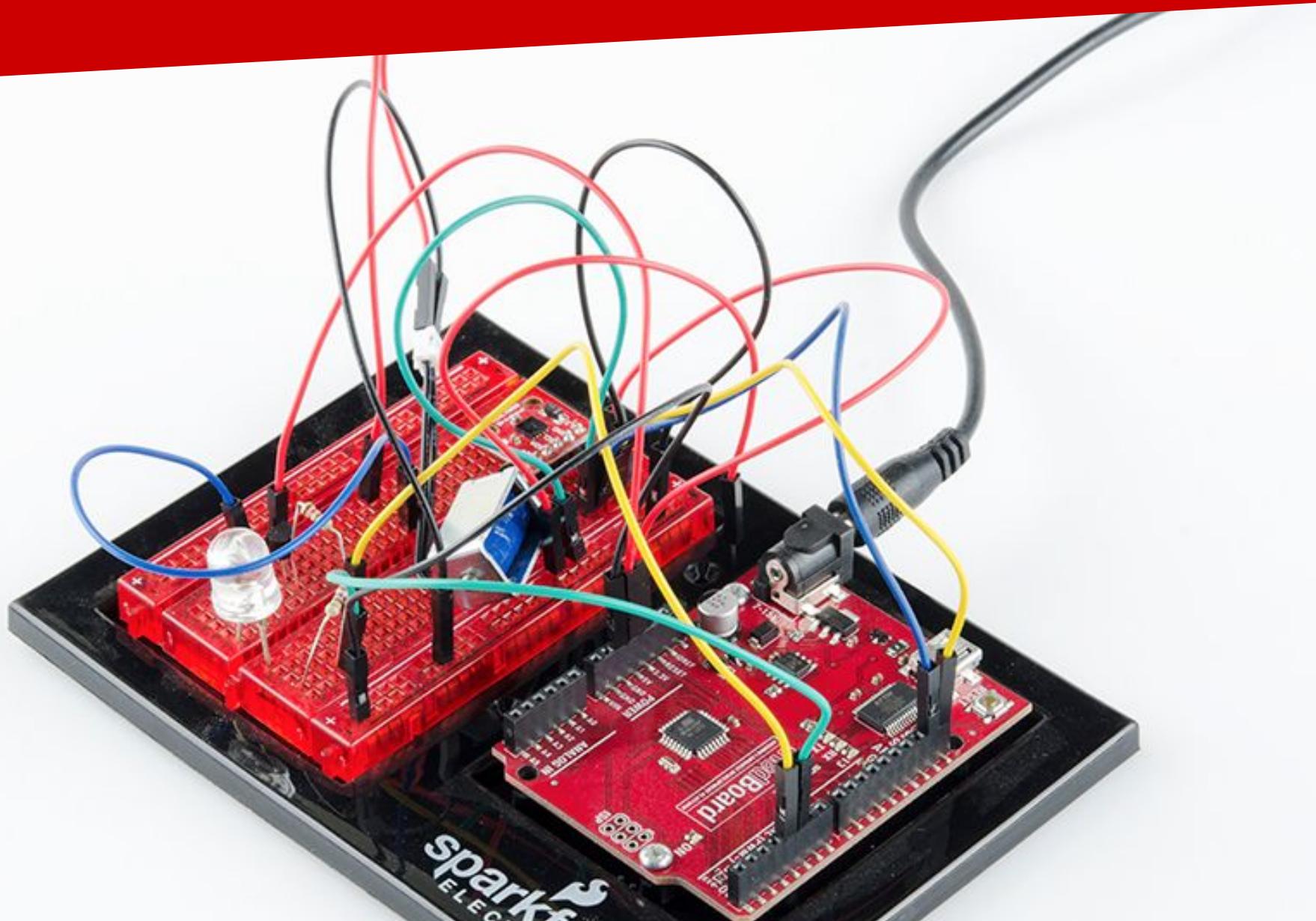


# Materials

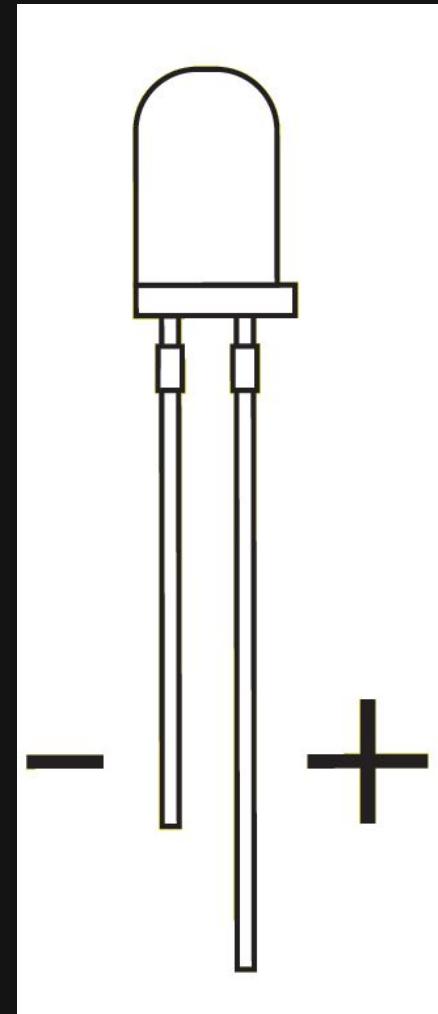
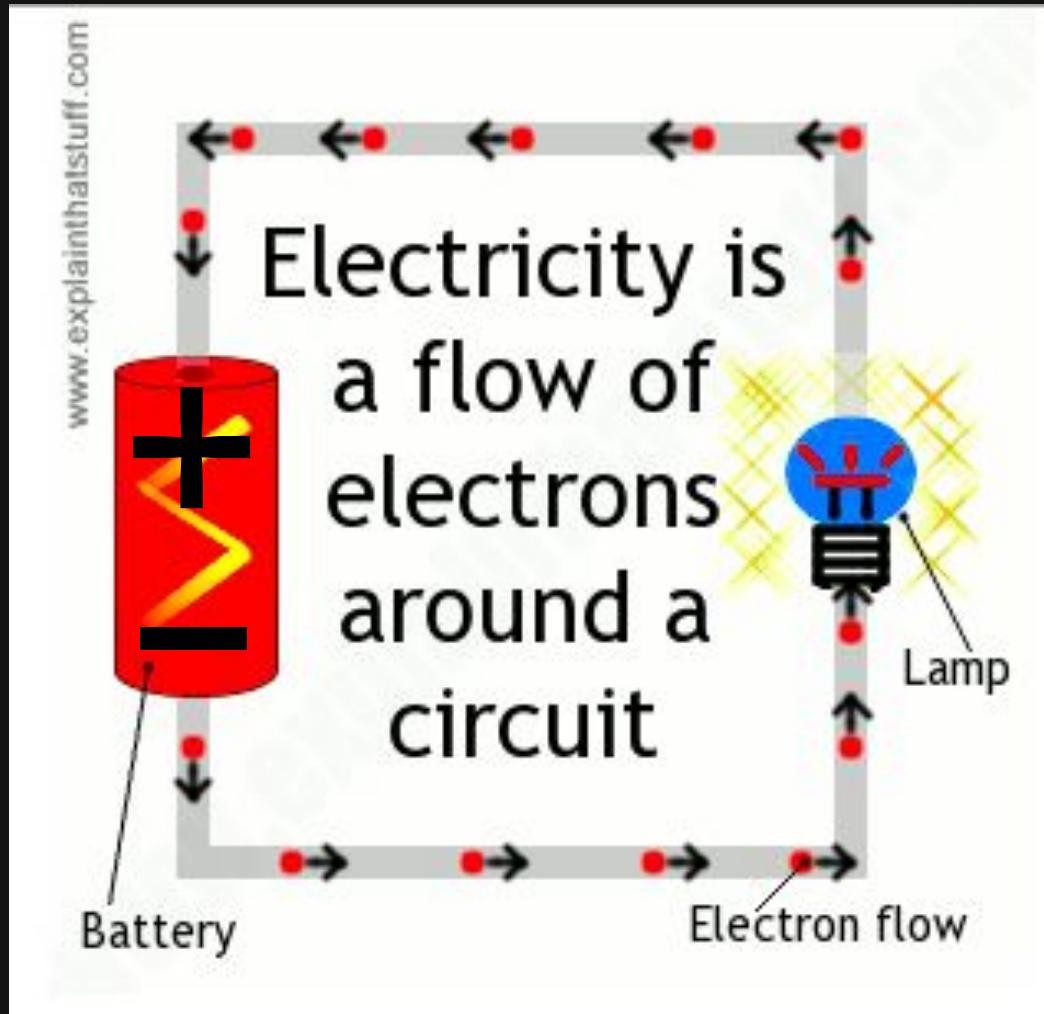


- Arduino
- Breadboard
- Components
- Jumper Wires

# Jumper Wires



# Circuit Basics



# Today you will...

- Connect Arduino;
- Prototype a circuit;
- Upload code;
- Customize code;
- Explore!

# Circuit #1

Connect an LED to pin 13

# Setup

- Connect Arduino to USB
- Settings - board & port

# Find...

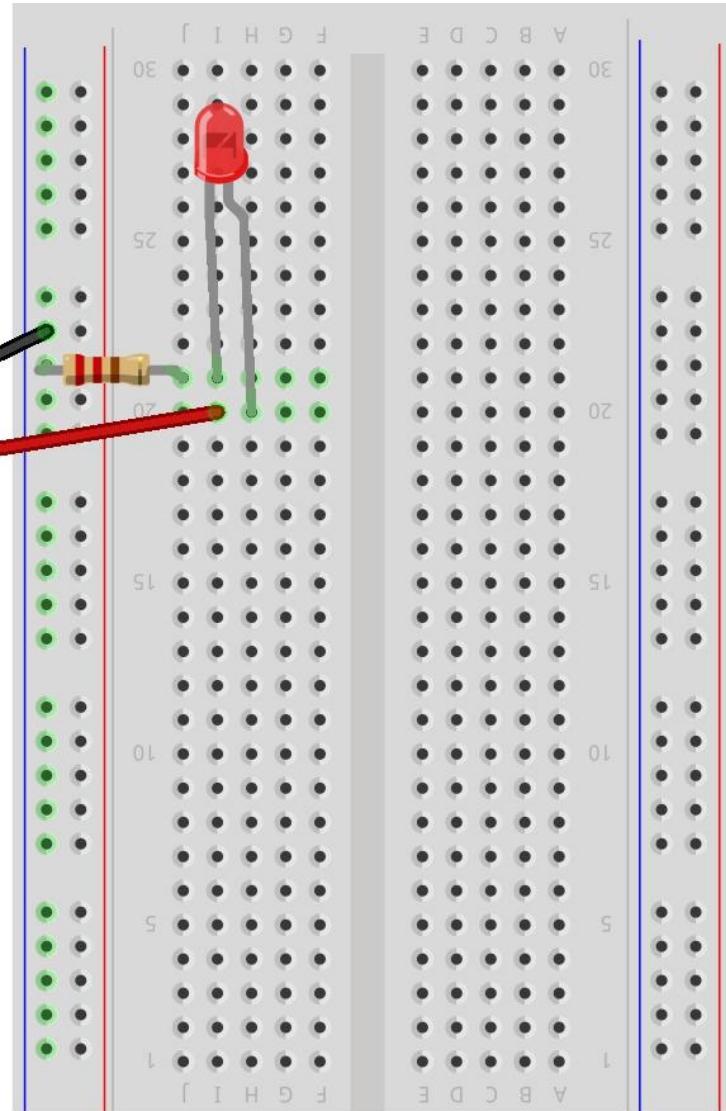
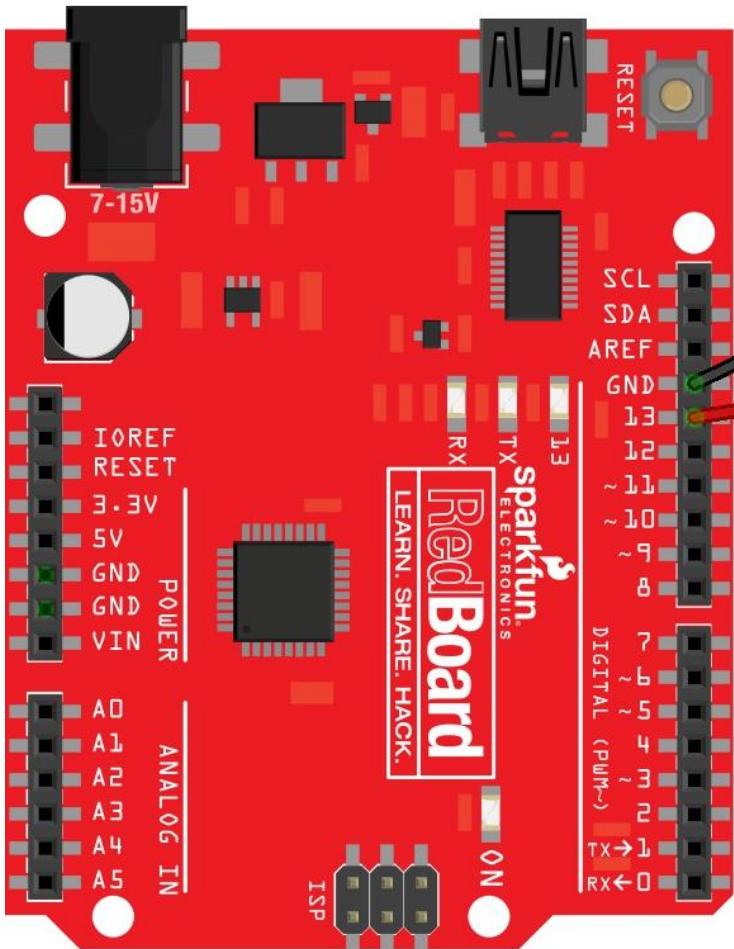


Short leg -  
Long leg +



Resistor  
(330ohm):  
limit current

# Build...

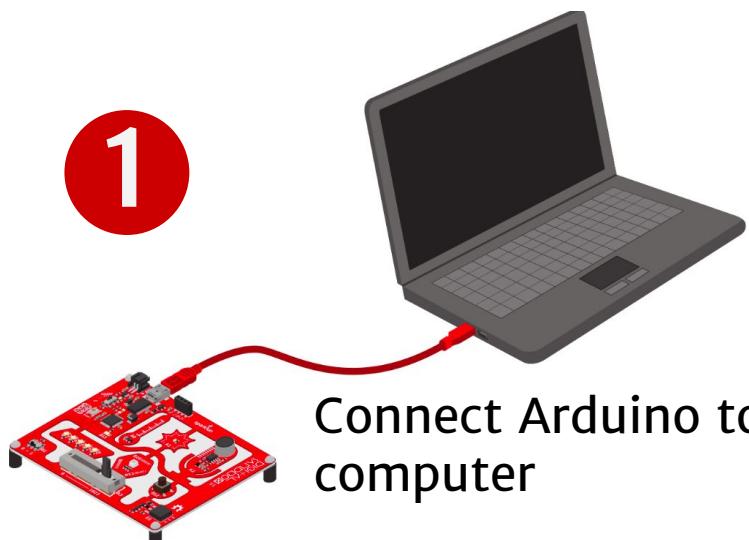


fritzing

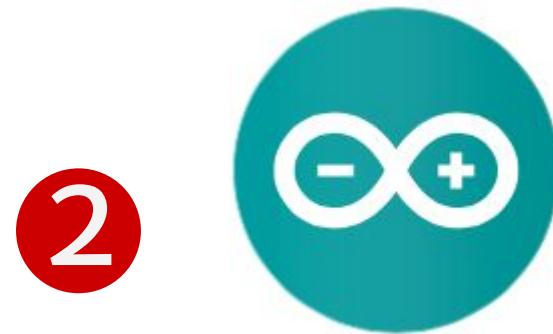
# Circuit #1

Get the LED to blink.

# Connect...



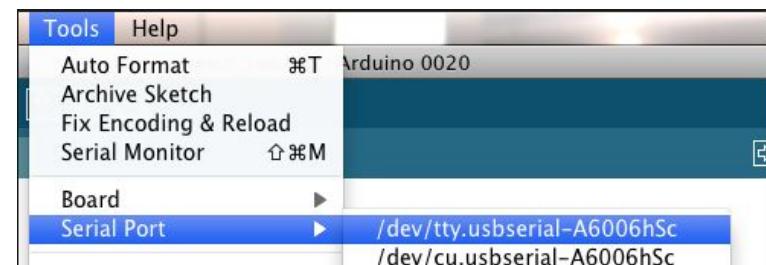
Connect Arduino to computer



Open Arduino software.



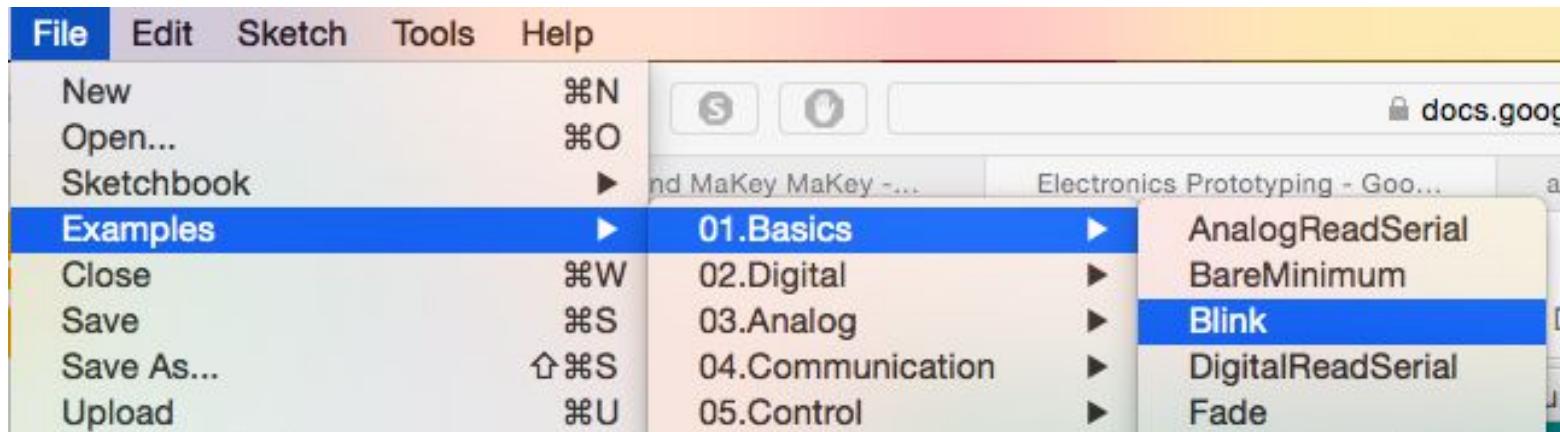
Set Board to "Arduino Uno"



Set Serial Port to "USB"

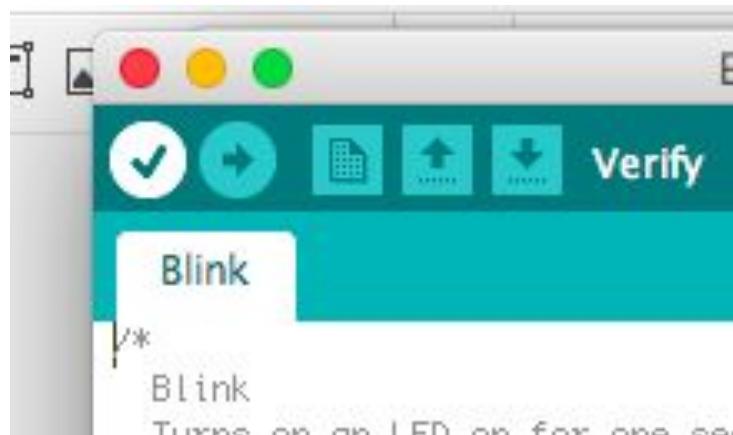
# Code...

1



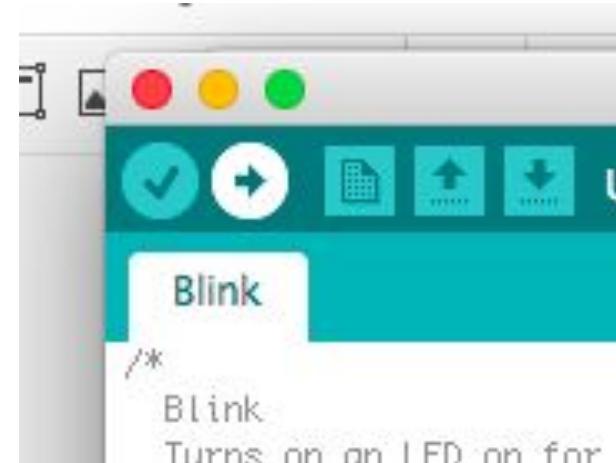
Load the example sketch “Blink.”

2



Verify the code.

3

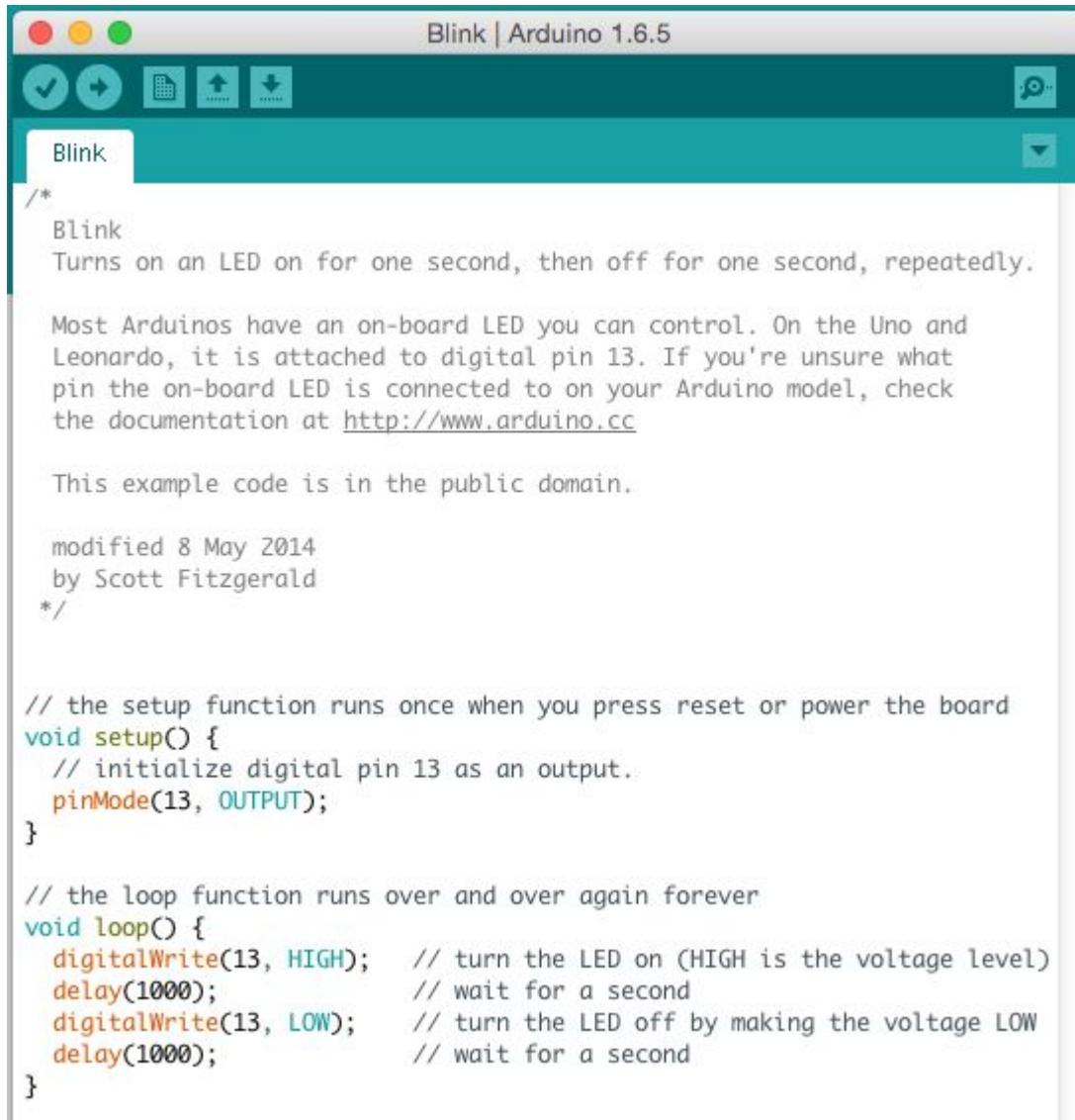


Upload to the board.

# Challenge #1

Get the LED to blink  
every 2.5 seconds.

# Customize...



The image shows a screenshot of the Arduino IDE version 1.6.5. The title bar reads "Blink | Arduino 1.6.5". The main window displays the "Blink" example sketch. The code is as follows:

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://www.arduino.cc

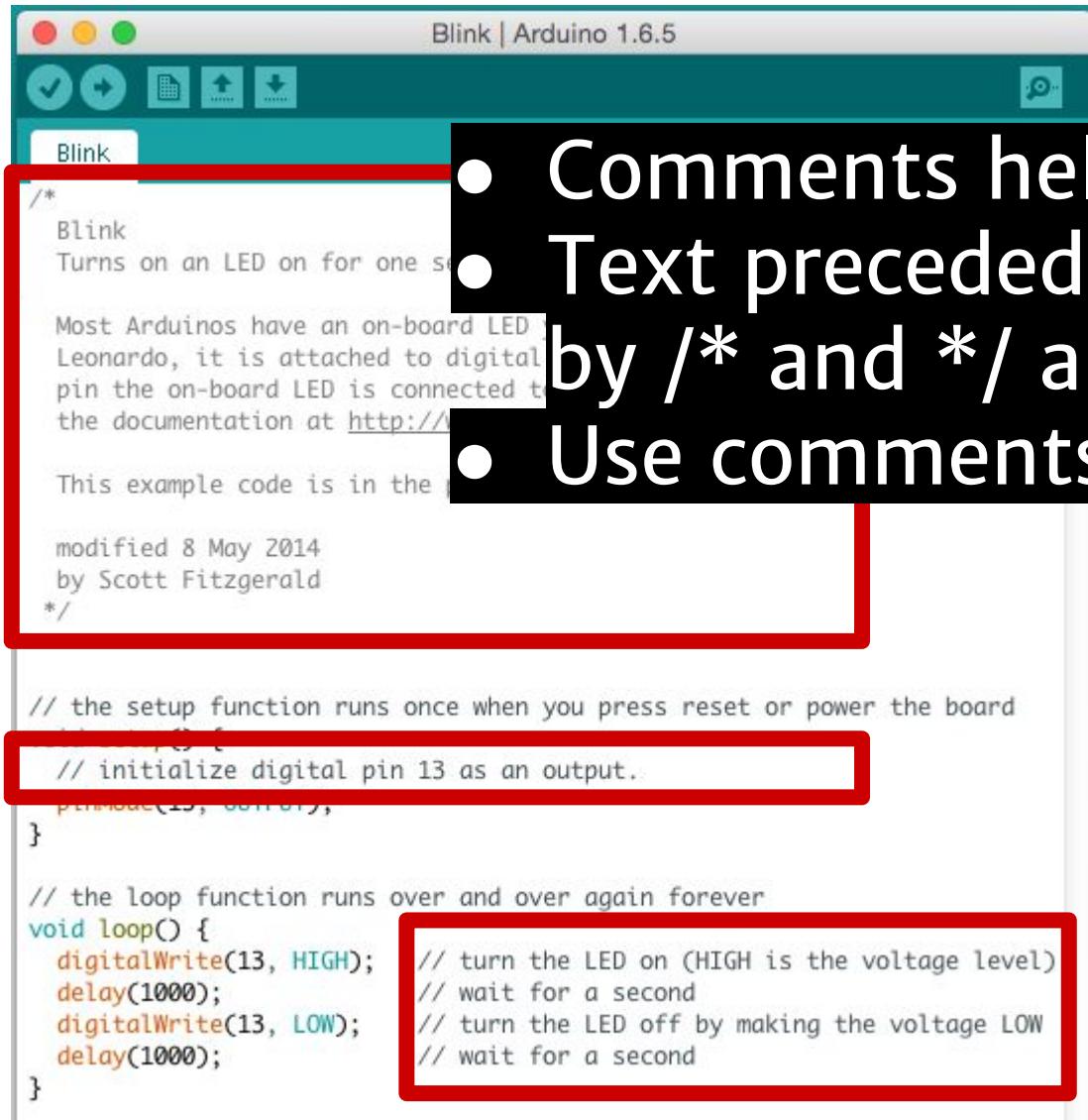
This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

# Customize...



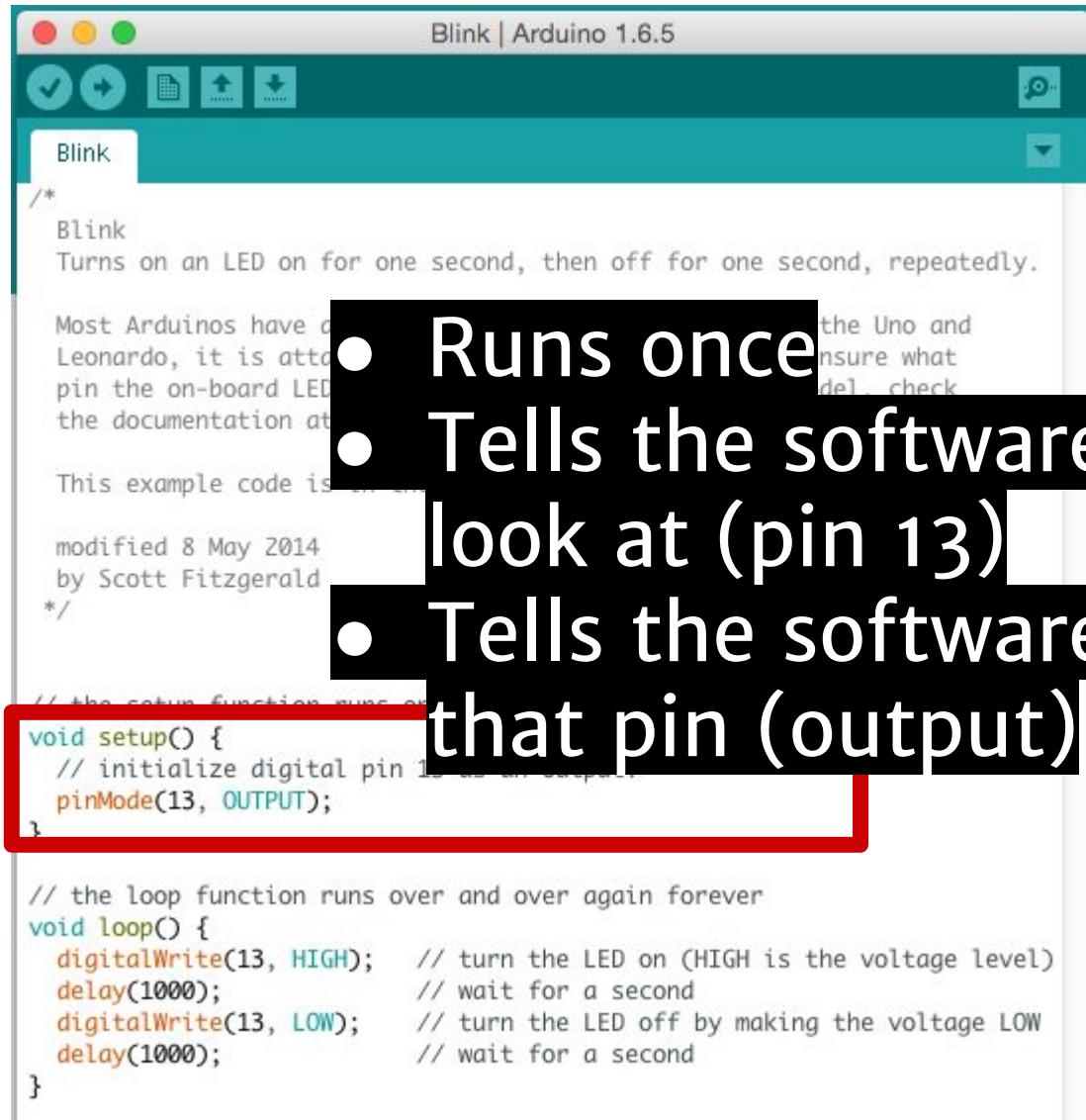
The image shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.5". The code editor displays the "Blink" example sketch. A red box highlights the explanatory text at the top of the code, which includes a summary of the sketch's purpose, notes about the LED connection, and a link to documentation. Another red box highlights the setup function, specifically the line that initializes digital pin 13 as an output. A third red box highlights the loop function, showing the code for turning the LED on and off with comments explaining each step.

```
/*
Blink
Turns on an LED on for one second, then turns it off
for one second, repeating this process over and over again.
Most Arduinos have an on-board LED. On the Arduino
Leonardo, it is attached to digital pin 13. The digital
pin the on-board LED is connected to is controlled by
the digitalWrite() command. For more information see
the documentation at http://www.arduino.cc/en/Tutorial/BlinkWithoutLibraries
This example code is in the public domain.
modified 8 May 2014
by Scott Fitzgerald
*/
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

- Comments help explain code
- Text preceded by // or bounded by /\* and \*/ are comments
- Use comments when you code

# Customize...



The image shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.5". The code editor displays the "Blink" example sketch. The code is as follows:

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have a
Leonardo, it is attached to
pin the on-board LED
the documentation at

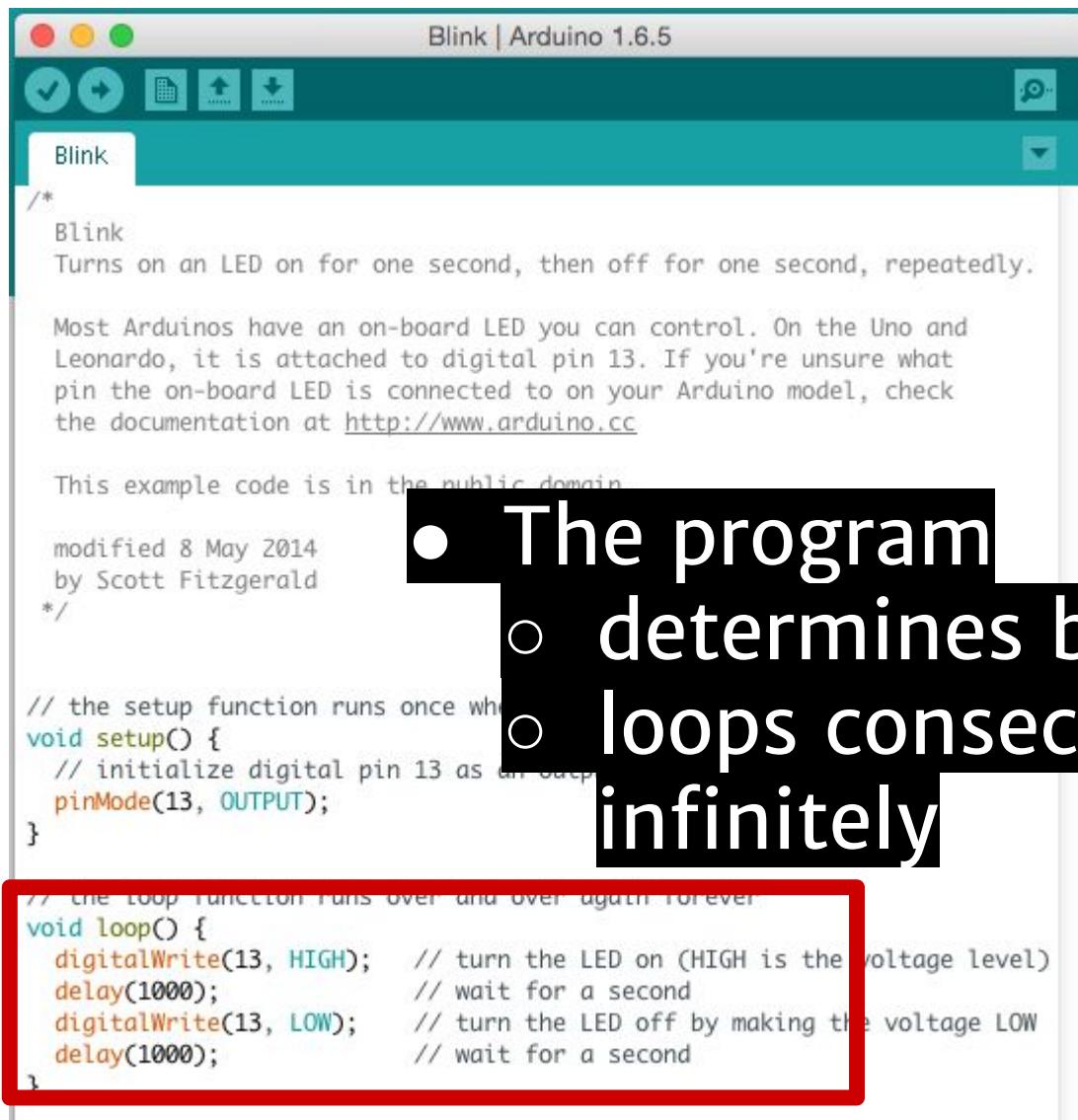
This example code is
modified 8 May 2014
by Scott Fitzgerald
*/
// the setup function runs once
void setup() {
  // initialize digital pin 13 as an output
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

A red rectangular box highlights the line of code `pinMode(13, OUTPUT);` in the `setup()` function.

- Runs once
- Tells the software which pin to look at (pin 13)
  - Tells the software how to treat that pin (output)

# Customize...



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.5". The code editor displays the "Blink" example sketch. The code is as follows:

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the Uno and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at http://www.arduino.cc

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/
// the setup function runs once when the sketch starts
void setup() {
  // initialize digital pin 13 as an output
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

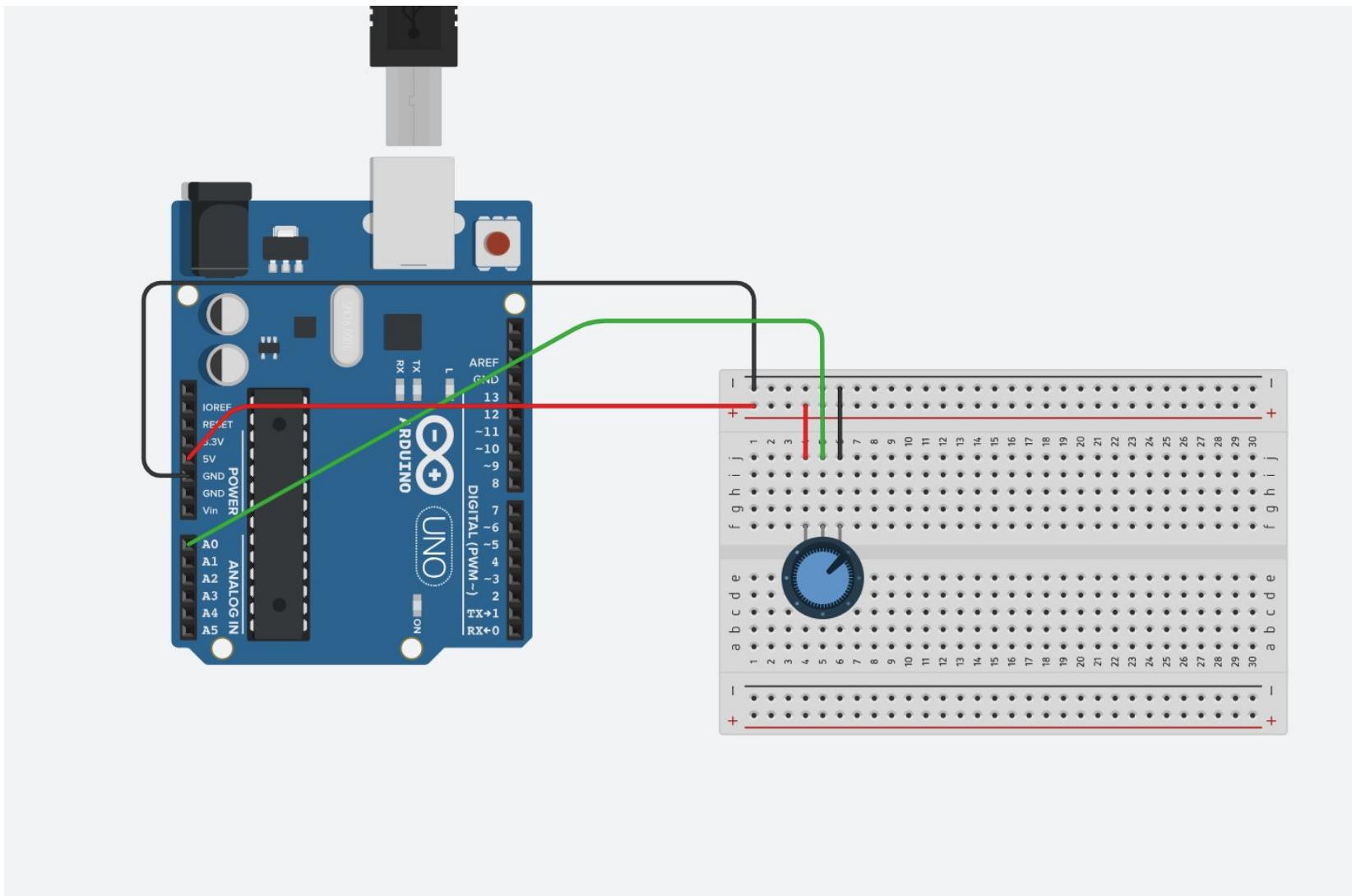
A large black rectangular box covers the middle portion of the code editor, containing the following text:

- The program
  - determines behaviours
  - loops consecutively and infinitely

# Circuit #2

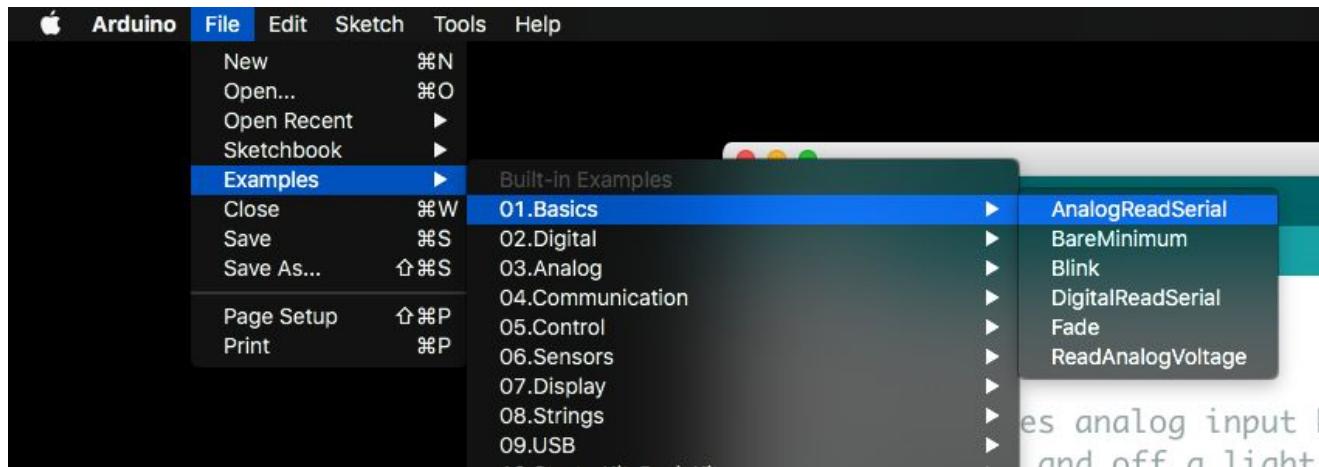
Connect an  
potentiometer to pin 5v,  
Ground, and Ao

# Build...



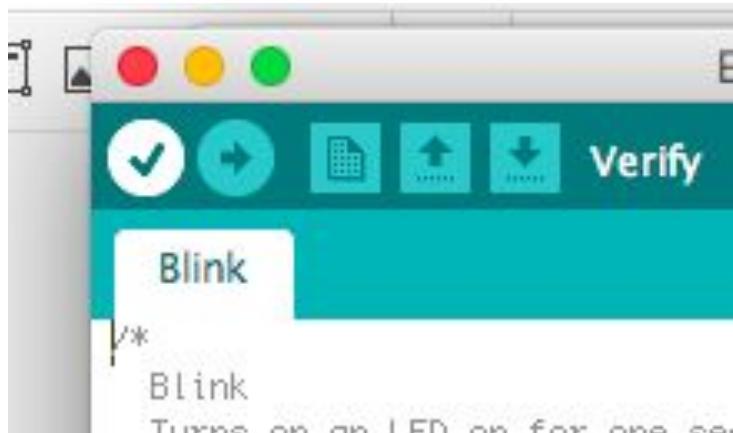
# Code...

1



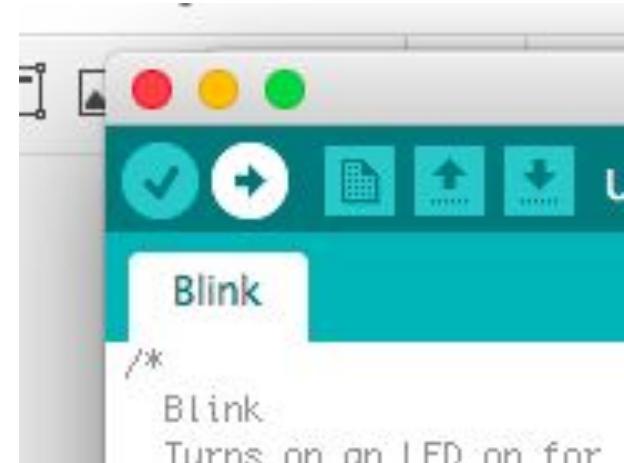
Load the example sketch “AnalogReadSerial.”

2



Verify the code.

3



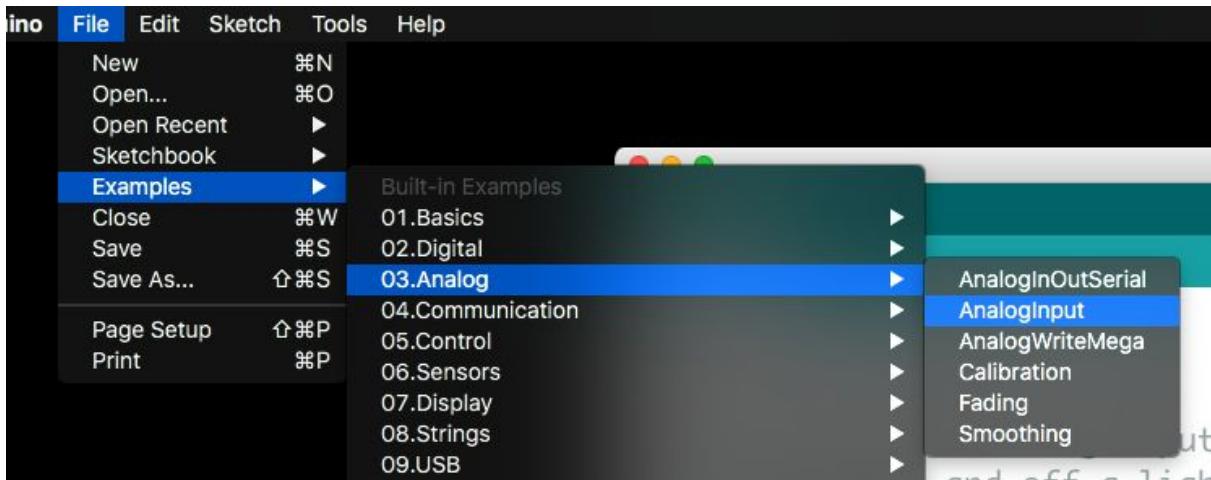
Upload to the board.

# Circuit #3

Combine Both Circuits

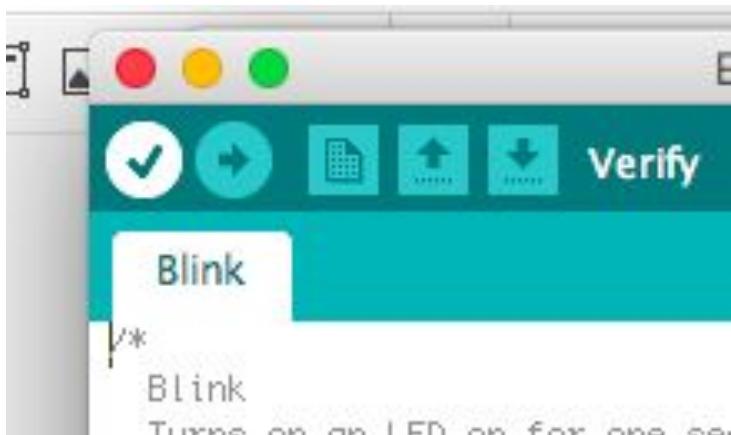
# Code...

1



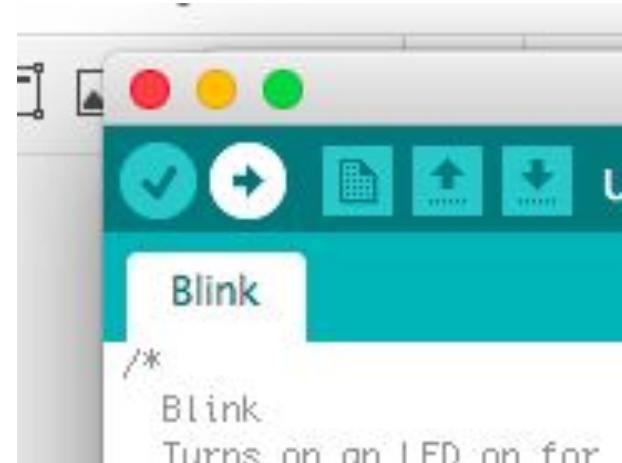
Load the example sketch “AnalogInput.”

2



Verify the code.

3



Upload to the board.

# Explore!

# Next Steps



**instructables**

let's make



Explore

Publish

Login | Sign Up

share what you make

Featured:

Intel IoT

Life Hacks

Beauty

Woodworking

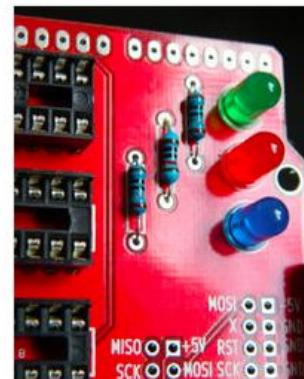
Let's Make

arduino



Arduino

by Instructables Guides in Arduino



Arduino

by playmobilley in Arduino



Arduino

by anouskadg in Arduino



ARDUINO  
XYLOPHONE

**instructables.com**

HOME / SEARCH RESULTS FOR ARDUINO

## 6,064 results for arduino

Confused about Arduino? Check out our [Arduino Buying Guide!](#)

PRODUCTS 802 TUTORIALS 364 BLOG 428 RESOURCES 43 CLASSES 0 ARTICLES 22 WISH LISTS 4403

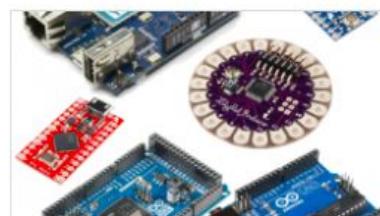
Tutorials 268 found



[What is an Arduino?](#)

FEBRUARY 26, 2013

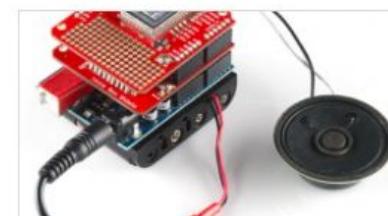
What is this 'Arduino' thing anyway?



[Arduino Comparison Guide](#)

NOVEMBER 11, 2013

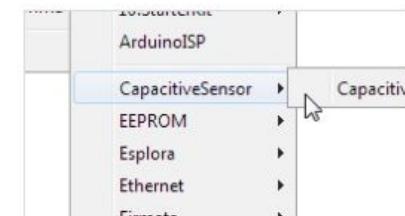
Examining the diverse world of Arduino boards.



[Arduino Shields](#)

AUGUST 6, 2013

All things Arduino Shields. What they are and how to assemble them.



[Installing an Arduino Library](#)

JANUARY 11, 2013

How do I install a custom Arduino library? It's easy!

# learn.sparkfun.com

learn.adafruit.com

Sign In    0 Items

# LEARN ARDUINO

32 GUIDES | 229 PAGES | 2 FEATURED | 6 POPULAR

`>> a = arduino('COM3');`  
`>> configureDigitalPin(a,9,'output');`

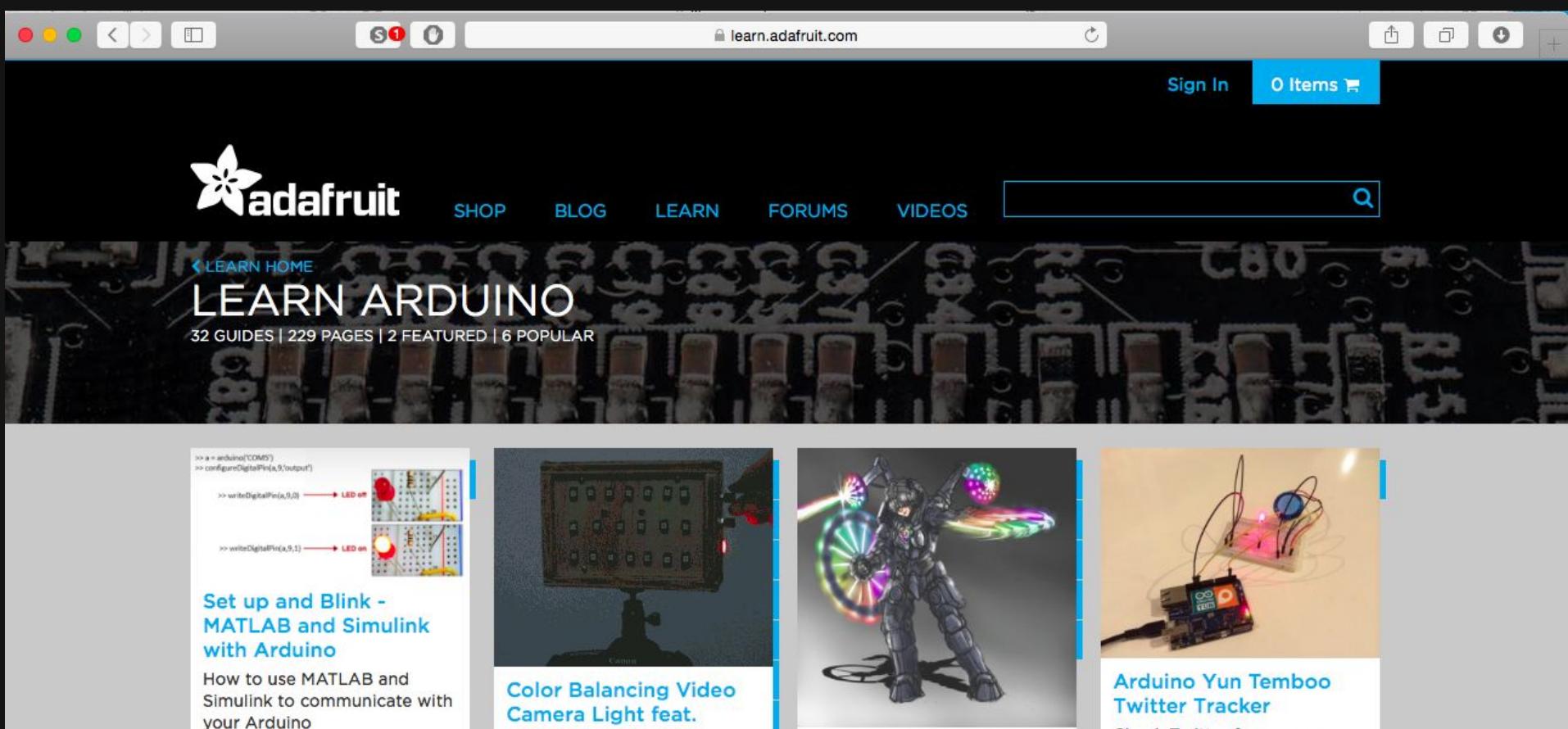
`>> writeDigitalPin(a,9,0) → LED off`

`>> writeDigitalPin(a,9,1) → LED on`

**Set up and Blink - MATLAB and Simulink with Arduino**  
How to use MATLAB and Simulink to communicate with your Arduino

**Color Balancing Video Camera Light feat.**

**Arduino Yun Temboo Twitter Tracker**



# learn.adafruit.com



<https://www.lib.ncsu.edu/techlending/makerspace>

## Makerspace



Digilent Analog Discovery 2  
The Digilent Analog Discovery 2 is a multi-function instrument that allows users to...  
*Walk Up,*  
*4 Hour*



SparkFun Arduino Inventor Kit  
Note: We have added an ESP8266 WiFi Shield to all of our kits! This enables WiFi-...  
*Walk Up,*  
*1 Week*



JoyLabz MaKey MaKey  
MaKey MaKey is a simple Invention Kit for beginners or experts doing art, engineering,...  
*Walk Up,*  
*1 Week*



Matter and Form 3D Scanner  
A quick, high quality 3D scanner for smaller objects  
Uses Matter and Form Software...  
*Walk Up,*  
*1 Week*



Internet of Things Starter Kit  
This Internet of Things (IoT) Starter Kit features the low-cost Adafruit Feather HUZZAH...  
*Walk Up,*  
*1 Week*



Occipital Structure 3D Scanner  
Senses 3D data for many applications. Checks out with an iPad Air 2 for handheld 3D...  
*Walk Up,*  
*1 Week*



Sphero SPRK Robotic Ball  
The Sphero SPRK Edition is Sphero's robotic ball platform designed to give orders to a...  
*Walk Up,*  
*1 Week*



iRobot Create 2 Programmable Robot  
Create 2 has a variety of programming methods to grasp the fundamentals of robotics,...  
*Walk Up,*  
*1 Week*



NeuroSky MindWave Mobile Brainwave Starter Kit  
Learn about how your brain responds to various stimulus



Moog Werkstatt-01 Analog Synthesizer  
The Moog Werkstatt-01 is a patchable, completely analog



Muse Brain Sensing Headband  
This brain sensing headband is a brain fitness tool that



littleBits Premium Kit  
The littleBits Premium Kit encourages you to learn the basics of electronics by...

# Ask Us.

[crnickel@ncsu.edu](mailto:crnickel@ncsu.edu)